

UNIVERSITÀ DI PISA



Facoltà di Scienze Matematiche, Fisiche e Naturali  
Corso di Laurea Specialistica in Informatica

TESI DI LAUREA

**PROGRAMMAZIONE LOGICA E BIOCHIMICA:  
ANALISI DI REFLUI**

RELATORI:

Prof. Paolo MANCARELLA

Prof. Roberto BARBUTI

CONTRORELATORE:

Prof. Fabio GADDUCCI

CANDIDATA:

Elisa Croci

Anno Accademico 2007-08

*A mio nonno Romano... , grazie!*

## Sommario

L'inquinamento ambientale è una delle problematiche più importanti perché la vita degli organismi è in pericolo in presenza di un ambiente contaminato.

Il monitoraggio biologico ha lo scopo di tenere sotto controllo il livello di inquinamento studiando le variazioni biologiche che avvengono negli organismi in relazione alla loro esposizione a sostanze tossiche.

Tramite il rilevamento dell'attività enzimatica nei reflui è possibile valutare il livello tossicologico di un refluo per l'ambiente.

L'utilizzo della programmazione logica ci ha permesso di descrivere il dominio dei reflui e, partendo dai dati di laboratorio, di indurre la loro composizione individuando le sostanze tossiche responsabili dell'impatto sull'attività enzimatica.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
<b>2</b>	<b>Gli enzimi: indicatori biologici di inquinamento</b>	<b>9</b>
2.1	I Bioindicatori . . . . .	10
2.1.1	I Biomarcatori . . . . .	14
2.2	L'attività enzimatica come biomarcatore . . . . .	15
2.2.1	Ruolo e definizione degli enzimi . . . . .	15
2.2.2	Inibizione dell'attività enzimatica come biomarcatore . . . . .	20
<b>3</b>	<b>Informatica e biologia</b>	<b>22</b>
3.1	Programmazione Logica . . . . .	23
3.1.1	Prolog . . . . .	24
3.1.2	Applicazioni del Prolog . . . . .	29
<b>4</b>	<b>Descrizione di “<i>EnzimiLab</i>”</b>	<b>31</b>
4.1	Il nucleo dell'applicazione . . . . .	34
4.1.1	Struttura del modello logico . . . . .	35
4.1.2	L'algoritmo per il calcolo della composizione del refluo . . . . .	37
4.2	Database . . . . .	47
4.3	Interfaccia grafica . . . . .	52
4.3.1	Collegamento con l'applicazione Prolog . . . . .	61

4.3.2 Collegamento con Microsoft Excel . . . . .	68
<b>5 Conclusioni e sviluppi futuri</b>	<b>70</b>

# Elenco delle figure

2.1	Passi base di una reazione biochimica . . . . .	17
2.2	Inibizione competitiva . . . . .	19
2.3	Inibizione non competitiva . . . . .	20
3.1	Esempio di programma Prolog . . . . .	25
3.2	Albero di ricerca per la richiesta <i>?- antenato(michele,teresa).</i>	27
4.1	Valori delle concentrazioni limite per legge dei metalli. . . . .	32
4.2	Rappresentazione degli esperimenti . . . . .	34
4.3	Rappresentazione di un reflu . . . . .	36
4.4	Rappresentazione dell'attività di un enzima . . . . .	40
4.5	Effettore candidato . . . . .	41
4.6	Effettori singoli . . . . .	43
4.7	Ammanco di attività enzimatica . . . . .	45
4.8	Selezione delle combinazioni additive . . . . .	46
4.9	Schema del database di EnzimiLab. . . . .	48
4.10	Mappa esperimenti . . . . .	53
4.11	Eliminazione di un enzima . . . . .	54
4.12	Selezione dei dati sperimentali da visualizzare . . . . .	55
4.13	Aggiunta dati sperimentali . . . . .	57
4.14	La visualizzazione dei reflui come albero. . . . .	58

---

4.15	Calcolo della composizione del refluo . . . . .	59
4.16	Eliminazione reflui . . . . .	60
4.17	Esportazione dei dati . . . . .	60

# Capitolo 1

## Introduzione

Le attività sociali, produttive e ricreative, principalmente in ambito urbano, richiedono ed utilizzano una grande quantità di acqua. La conseguenza diretta dell'utilizzo dell'acqua è la produzione di scarichi che, per poter essere restituiti all'ambiente, devono necessariamente essere sottoposti ad un trattamento depurativo.

Le *acque reflue urbane*, che in passato contenevano quasi esclusivamente sostanze biodegradabili, presentano attualmente maggiori problemi di smaltimento a causa della presenza sempre più ampia di composti chimici di origine sintetica, impiegati prevalentemente nel settore industriale.

Il mare, i fiumi ed i laghi non sono in grado di ricevere una quantità di sostanze inquinanti superiore alla propria capacità autodepurativa senza vedere compromessa la qualità delle proprie acque ed i normali equilibri dell'ecosistema. È evidente quindi la necessità di depurare le acque reflue.

Il trattamento del refluo è tanto più spinto quanto più i corpi idrici recettori (mari, fiumi, laghi, ...) risultano a rischio di inquinamento permanente.

Una corretta gestione ambientale delle acque è strettamente legata all'identificazione della presenza di composti inquinanti e alla conoscenza dello



stato ecologico dell'ambiente di studio.

L'analisi dell'inquinamento delle acque si basa sul monitoraggio chimico-fisico e sul biomonitoraggio. Il monitoraggio chimico-fisico si occupa dell'analisi delle sostanze pericolose (solventi, idrocarburi, pesticidi, metalli pesanti), dotate di proprietà tossiche una volta in contatto con gli organismi viventi, in quanto ne danneggiano strutture e funzioni, provocandone in alcuni casi la morte, e che possono interagire con l'uomo attraverso l'acqua potabile o l'esposizione durante attività ricreative. Tramite il monitoraggio biologico (biomonitoraggio) si studia lo stato di salute ambientale mediante l'osservazione dei parametri vitali degli organismi che vivono nell'ambiente studiato.

In questo lavoro abbiamo selezionato come indicatore del livello di inquinamento di un reffuo *il livello dell'attività enzimatica* di un insieme di enzimi perché quest'ultimi ricoprono un ruolo cruciale nella sopravvivenza degli essere viventi e se la loro attività è compromessa lo è anche la nostra vita.

Il gruppo di ricerca appartenente al Dipartimento di Biologia dell'Università di Pisa, guidato dal professore Umberto Mura, ha selezionato un insieme di enzimi che svolgono ruoli diversi all'interno del metabolismo degli esseri viventi e ha effettuato una serie di esperimenti volti a calcolare il livello dell'attività enzimatica di ciascun enzima in presenza di sostanze tossiche organiche (benzene, cloroformio, fenolo, ...) e inorganiche (argento, ferro, rame, ...).

Una volta fatti tutti questi esperimenti, come servirsi della grande quantità di dati prodotta? Se ho un campione di reffuo e rilevo il livello di attività enzimatica degli enzimi in relazione ad esso come posso risalire a quali sostanze tossiche compongono il reffuo?

L'applicazione sviluppata in questo lavoro, "*EnzimiLab*", è stata sviluppata per agevolare il calcolo della composizione di un reflu.

Il nucleo di questa applicazione è stato sviluppato in Prolog in quanto essendo un linguaggio dichiarativo, basato sulla logica, ci ha permesso di descrivere in modo semplice ma allo stesso tempo formale, le relazioni che intercorrono tra gli enzimi e le sostanze tossiche e di dedurre partendo da esse le possibili composizioni di un reflu. Inoltre la programmazione logica permette una gestione semplice di vincoli e l'abduzione, utili per uno sviluppo futuro dell'applicazione quando saranno disponibili maggiori informazioni sul dominio analizzato.

I dati degli esperimenti di laboratorio effettuati dai biologi sono stati raccolti, organizzati e inseriti in una base di dati alla quale è possibile accedere mediante l'interfaccia utente. Questi dati vengono utilizzati dal nucleo Prolog per dedurre la composizione del reflu.

## Sintesi

Il *capitolo 2* spiega il ruolo dei bioindicatori e biomarcatori nel monitoraggio dell'inquinamento, cosa sono gli enzimi e perchè sono importanti per la sopravvivenza degli esseri viventi.

Il *capitolo 3* mostra i legami profondi che esistono tra la biologia e l'informatica. Introduce, in modo informale, il linguaggio Prolog e le motivazioni per cui la programmazione logica è stata scelta per risolvere questo problema.

Nel *capitolo 4* viene presentata l'applicazione *EnzimiLab* partendo dal nucleo scritto in Prolog, descrivendo la base di dati adottata per strutturare la grande quantità di dati a disposizione e l'interfaccia utente sviluppata in C con l'ausilio delle librerie grafiche GTK+2 la quale permette, da un lato di

calcolare la composizione dei reflui interfacciandosi con il Prolog e dall'altro di gestire i dati: inserimento, cancellazione, modifica, esportazione in formato .csv e Microsoft Excel comunicando con la base di dati.

La scelta del linguaggio C è stata dettata dalla necessità di efficienza almeno a livello di interfaccia utente in quanto il nucleo Prolog è pesante e rallenta l'applicazione.

Nel *capitolo 5* infine vengono riassunti gli obiettivi raggiunti e le condizioni che permetterebbero un ulteriore sviluppo di *Enzimilab*.

## Capitolo 2

# Gli enzimi: indicatori biologici di inquinamento

Ogni settore della nostra vita (dallo sport al mondo degli affari, dalla valutazione dello stato di salute di persone a quella della qualità della vita ...) ha dei parametri per valutare la qualità del suo stato. In alcune attività sono ormai consolidati certi indicatori universali, come per esempio il Prodotto Interno Lordo delle nazioni usato in macroeconomia; in altre attività, gli indicatori scelti per controllare l'evoluzione di fenomeni sociali, fisici, o biologici sono ancora opinabili e oggetto di dibattito. Il monitoraggio ambientale è tra questi settori.

Nel settore del monitoraggio dell'inquinamento ambientale infatti si stanno sviluppando tecniche integrative alla raccolta sistematica di dati qualitativi e quantitativi fatta con una procedura standardizzata in un periodo di tempo necessario a raccogliere i dati previsti infatti negli ultimi anni è stato preso in considerazione il possibile utilizzo di sistemi biologici per il monitoraggio (*biomonitoraggio*) ambientale. Rispetto alle tecniche analitiche

tradizionali, nel biomonitoraggio si valuta lo stato di salute dell'ambiente studiando l'effettiva tossicità degli agenti inquinanti osservando le reazioni, a vari livelli (organismo, organo, cellula, ...), *negli organismi che vivono a contatto nell'ambiente analizzato*. Tale tecnica ha costi di gestione limitati e dà la possibilità di coprire con relativa facilità vaste zone e territori diversificati consentendo una adeguata mappatura del territorio.

Nel biomonitoraggio si possono distinguere due diverse tipologie di organismo test:

**‘Bioindicatori’:** organismi che subiscono variazioni evidenti nella fisiologia, nella morfologia o nella distribuzione sotto l'influsso delle sostanze presenti nell'ambiente

**‘Bioaccumulatori’:** organismi in grado di sopravvivere in presenza di inquinanti che accumulano nei loro tessuti

Data l'importanza globale del tema dell'inquinamento, l'Unione Europea ha adottato una politica ambientale di *prevenzione* emanando una serie di direttive come ad esempio la Direttiva 2000/60/CE del 23 Ottobre 2000 dove viene data la definizione di “valori limite di emissione” come: *la massa espressa in rapporto a determinati parametri specifici, la concentrazione e/o il livello di un'emissione che non devono essere superati in uno o più periodi di tempo*. Nella Tabella 2.1 a titolo esemplificativo vengono elencati i limiti di legge per le concentrazioni dei metalli nei reflui (questi limiti di legge sono stati definiti anche per altri tipi di composti, ad esempio composti organici).

## 2.1 I Bioindicatori

In ecosistemi inquinati le sostanze tossiche entrano continuamente in contatto con organismi viventi alterando il loro stato di salute. Inizialmente un

Valori delle concentrazioni limite per legge		
Metallo	mg/l	[ $\mu$ M]
Argento	1	37
Bario	20	146,5
Boro	4	370
Cadmio	0,1	0,89
Cobalto	2	34
Cromo	2	38,5
Ferro	2	35,8
Fluoro	1,5	79
Magnesio	50	2057
Manganese	2	36,4
Mercurio	0,005	0,025
Molibdeno	1	10,4
Nichel	2	34,1
Piombo	0,5	2,4
Rame	1	15,7
Selenio	0,03	0,38
Stagno	10	84,2
Vanadio	0,1	19,6
Zinco	2	30,6

Tabella 2.1: Valori delle concentrazioni limite per legge dei metalli.

organismo cerca di proteggersi convertendo le sostanze nocive, di solito poco solubili, in forme maggiormente idrosolubili in modo da poterle facilmente espellere mediante urina o sudore, oppure tali sostanze possono essere immagazzinate in alcuni tessuti oppure possono entrar a far parte di complessi con alcune proteine. I danni causati dall'assorbimento di sostanze inquinanti dipendono non solo dalla sostanza tossica in questione, ma anche dal tipo di organismo analizzato. Sicuramente queste sostanze provocano una "sindrome da stress" nell'organismo, cioè un'alterazione *misurabile* del suo stato fisiologico indotta da un cambiamento ambientale. La sindrome da stress può

essere facilmente quantificata mediante l'utilizzo di opportuni indici, detti anche *indicatori biologici*.

Una definizione di indicatore biologico fu proposta da Iserentant e De Sloover nel 1976:

*“ Un indicatore biologico è un organismo o un sistema biologico usato per valutare una modificazione, generalmente degenerativa, della qualità dell'ambiente, qualunque sia il suo livello di organizzazione e l'uso che se ne fa. Secondo i casi il bioindicatore sarà una comunità, un gruppo di specie con comportamento analogo (gruppo ecologico), una specie particolarmente sensibile (specie indicatrice), oppure una porzione di organismo, come organi tessuti cellule o anche una soluzione di estratti enzimatici”*

Come si può capire da questa definizione, un bioindicatore non ha una natura ben precisa. Si può identificare come bioindicatore una popolazione, un organismo, un organo ma è anche possibile scendere più in dettaglio fino al livello cellulare e biochimico di un organismo vivente, a seconda dell'ambiente in cui si intende rilevare delle alterazioni. Alcuni esempi di bioindicatori sono: licheni e muschi (inquinamento atmosferico), invertebrati acquatici (inquinamento acque dolci), mammiferi ed uccelli (indicatori di disturbo antropico), specie fossili (indicatori paleoecologici) ...

Comunque, indipendentemente dalla sua natura, un bioindicatore deve avere i seguenti requisiti:

**1. Accessibilità:**

- deve essere facilmente campionabile
- deve essere misurabile con tecniche facili

**2. Idoneità bio-ecologica:**

- deve avere un'ampia distribuzione nell'area di studio
  - deve essere di facile identificazione ed è necessario avere adeguate conoscenze su anatomia, fisiologia ed ecologia dell'indicatore
3. **Lungo ciclo vitale:** in questo modo l'indicatore ha una maggiore probabilità di stare a contatto con l'inquinante
4. **Scarsa mobilità e facile reperibilità in tutte le stagioni**
5. **Affidabilità:** deve fornire la stessa risposta in situazioni identiche, nello spazio e nel tempo
6. **Rappresentatività:**
- deve essere chiaramente correlabile con un certo fenomeno o una certa caratteristica che si vuole rilevare o controllare
  - deve essere altamente correlabile con l'effetto suddetto, con un minimo di dispersione statistica
  - deve essere difficilmente camuffabile da fattori di disturbo
  - deve avere una validità sufficientemente generalizzabile a molte situazioni analoghe, anche se non identiche
  - deve essere facilmente misurabile e possibilmente monitorabile automaticamente

Una tipologia importante di bioindicatore frutto della recente evoluzione delle tecniche di biologia molecolare è il **biomarcatore**.



### 2.1.1 I Biomarcatori

La National Academy of Science statunitense definisce biomarcatore “*quella variazione, indotta da un contaminante, a livello delle componenti biochimiche o cellulari, che può essere misurata in un sistema biologico*”.

Il concetto si è evoluto e oggi il biomarcatore può essere definito come “*la variazione di una risposta biologica (biochimica, molecolare, cellulare, fisiologia o comportamentale) che può essere correlata all’esposizione e/o all’effetto tossico di composti inquinanti*” (Peakal et al., 1994). Esso può includere ogni tipo di parametro misurabile nei fluidi biologici, in cellule, tessuti o animali, che riflette un’interazione tra un sistema biologico e un potenziale insulto chimico, fisico, biologico.

I biomarcatori possono essere suddivisi in tre classi:

**Biomarcatore d’esposizione:** la risposta di un organismo che indica l’esposizione a un composto chimico o una classe di composti, ma che non fornisce nessuna indicazione dei reali effetti tossicologici sull’organismo

**Biomarcatore d’effetto:** è una modificazione biochimica o fisiologica, misurabile in un tessuto o in un fluido corporeo, associata ad un possibile o effettivo stato di malattia

**Biomarcatore di suscettibilità:** si riferisce all’innata o all’acquisita capacità di un organismo a rispondere ad una specifica sostanza xenobiotica<sup>1</sup>

---

<sup>1</sup>Con il termine *xenobiotico* si definisce una sostanza chimica che è estranea al sistema biologico. La categoria include i farmaci, i contaminanti ambientali, gli agenti cancerogeni, gli insetticidi, ma anche composti di origine naturale e composti che si originano per l’aggiunta di additivi chimici o in seguito alla cottura dei cibi. In rapporto alla loro natura e concentrazione, gli xenobiotici possono determinare effetti nocivi sull’uomo, sull’animale o in generale sugli ecosistemi.

I biomarcatori di esposizione rivelano la presenza di xenobiotici o dei loro metaboliti<sup>2</sup> quindi possono essere utilizzati per confermare o stimare l'esposizione di individui o popolazioni ad una particolare sostanza, fornendo un legame tra esposizione esterna e dose interna.

I biomarcatori di effetto valutano la risposta dell'organismo e possono essere utilizzati per documentare effetti avversi dovuti all'esposizione e all'assorbimento di una sostanza chimica. Generalmente i biomarcatori d'effetto più sensibili sono rappresentati da alterazioni nei livelli e nell'attività degli enzimi (induzione o inibizione di enzimi che metabolizzano gli xenobiotici).

## 2.2 L'attività enzimatica come biomarcatore

I biomarcatori enzimatici sono validi strumenti di indagine per identificare agenti tossici e/o inquinanti nell'ambiente. Prima di analizzare il loro ruolo come biomarcatori vediamo cosa sono gli enzimi e quale ruolo rivestono nella vita di un organismo.

### 2.2.1 Ruolo e definizione degli enzimi

#### Ruolo

Gli esseri viventi, mediante una catena di reazioni chimiche, identificabili come *cicli metabolici*<sup>3</sup>, riescono a sintetizzare i nutrienti che trovano nel loro ambiente in molecole essenziali per la loro sopravvivenza. Il problema di queste reazioni è la velocità a cui avvengono: una velocità troppo lenta, in-

---

<sup>2</sup>I *metaboliti* sono sostanze che prendono parte alle reazioni chimiche che avvengono nell'organismo oppure che derivano da esse.

<sup>3</sup>*Ciclo metabolico*: l'insieme delle reazioni metaboliche che riproduce ciclicamente le stesse sostanze (es. il ciclo di Krebs nella respirazione).

conciliabile con i tempi richiesti dalla vita.

Ad esempio, una reazione semplice come l'idratazione del biossido di carbonio è catalizzata da un enzima, detto anidrasi carbonica. In assenza di questo enzima il trasferimento di ossigeno dai tessuti al sangue e poi all'aria alveolare sarebbe meno efficiente.

Tutto questo sottolinea l'importanza della presenza degli enzimi per la sopravvivenza degli organismi e perciò è un valido biomarcatore nell'analisi dell'impatto dell'inquinamento sulla salute degli organismi che popolano l'ambiente.

### Definizione

Gli *enzimi*, i *catalizzatori*<sup>4</sup> dei sistemi biologici, sono importanti molecole che determinano tutte le trasformazioni chimiche cellulari. Quasi tutti gli enzimi conosciuti sono proteine, anche se recentemente si è scoperto che anche alcune molecole di RNA sono cataliticamente attive.

Le caratteristiche principali di un enzima sono:

**Potere catalitico:** quante volte riesce ad accelerare una determinata reazione.

La catalisi avviene in una regione particolare dell'enzima detta *sito attivo*.

**Specificità:** indica il tipo di reazione catalizzata e i reagenti, detti anche *substrati*, coinvolti. Solitamente un enzima catalizza una sola reazione chimica o una serie di reazioni strettamente correlate.

---

<sup>4</sup>Un catalizzatore è una sostanza, fonte o dispositivo che interviene in una reazione chimica aumentandone la velocità ma rimanendo inalterato al termine della stessa

**Saturabilità:** premettendo che la velocità di una reazione chimica dipende dalle concentrazioni dei substrati coinvolti, la saturabilità indica per quale concentrazione dei substrati la velocità della reazione è massima.

**Inibilità:** con *inibizione* di un enzima si intende un suo malfunzionamento, temporaneo o permanente, sia che diminuisca il suo potere catalitico, sia che ne diminuisca l'affinità con il suo substrato. Un enzima può essere inibito da diversi fattori e in modo differente a seconda del meccanismo d'azione dell'inibitore (una molecola che assomiglia molto al substrato per forma, distribuzione di cariche ...).

Il modello più semplice di una reazione catalizzata da un enzima è:

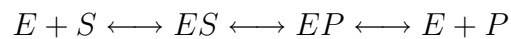


Figura 2.1: Passi base di una reazione biochimica

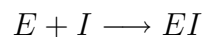
Analizzando la Figura. 2.1 vediamo che l'enzima, E, si combina con il substrato, S, per formare il complesso enzima-substrato, ES. Il complesso ES può andare incontro a due possibili destini: si può dissociare di nuovo in E e S oppure può procedere per formare a sua volta un complesso enzima-prodotto, EP. Il complesso EP può tornare indietro a formare il complesso ES oppure si può scindere in prodotto P ed enzima libero E che sarà così nuovamente

disponibile per reagire con un'altra molecola di substrato.

Gli *enzimi allosterici*<sup>5</sup>, oltre al sito attivo dove si lega il substrato, possiedono un altro sito dove si lega il regolatore (attivatore o inibitore). Il legame del regolatore modifica la conformazione del sito attivo e quindi modifica l'attività catalitica dell'enzima.

L'inibizione enzimatica è di due tipi:

**Irreversibile (o avvelenamento dell'enzima)** : è causata da alcune sostanze tossiche quali gas nervini, piombo, mercurio, . . . L'inibitore agisce sul sito attivo modificando permanentemente l'enzima, disattivandolo totalmente. Questo tipo di inibizione, illustrata nella Figura. 2.2, si può descrivere con la seguente equazione:



**Reversibile** : gli inibitori in questo caso si legano all'enzima mediante legami deboli che si formano e si distruggono molto rapidamente, perciò il loro effetto sull'enzima è istantaneo e non disattiva completamente l'enzima. L'inibizione reversibile si manifesta in tre modi:

- Nell'*inibizione competitiva* l'inibitore, può legarsi solamente all'enzima libero e non al complesso enzima-substrato, perciò il substrato e l'inibitore competono per lo stesso sito attivo sull'enzima (vedi figura 2.2). Un inibitore competitivo diminuisce la velocità di catalisi riducendo il numero di molecole enzimatiche accessibili al substrato. Per ogni concentrazione di

---

<sup>5</sup>*Allosterico* significa "altro sito" (oltre al sito attivo).

inibitore, [I], l'inibizione può essere rimossa da un aumento della concentrazione del substrato, [S].

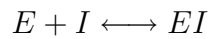


Figura 2.2: Inibizione competitiva - l'inibitore si sostituisce al substrato nel sito attivo dell'enzima

- Nell'*inibizione non competitiva* l'inibitore e il substrato si possono legare simultaneamente alla molecola enzimatica, ma in siti differenti formando così il complesso enzima-substrato-inibitore, EIS (vedi Figura. 2.3). La velocità della reazione diminuisce all'aumentare della concentrazione dell'inibitore, quindi questo tipo di inibizione è più marcata ad alte concentrazioni di substrato, in quanto aumentando la concentrazione di substrato, aumenta anche quella del complesso enzima-substrato, la forma alla quale l'inibitore si lega. Tuttavia il complesso EIS non procede verso la formazione del prodotto della reazione.



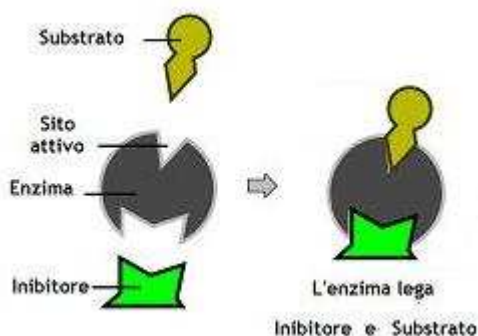


Figura 2.3: Inibizione non competitiva - l'inibitore impedisce che il substrato venga trasformato in prodotto

### 2.2.2 Inibizione dell'attività enzimatica come biomarcatore

In letteratura si trovano molti esempi riguardanti l'inibizione di enzimi da parte di agenti inquinanti organici e inorganici.

Tra i diversi biomarcatori, un posto di rilievo è occupato dall'attività acetilcolinesterasica (AChE)[7], utilizzata già da parecchi anni per valutare l'esposizione di alcuni organismi (soprattutto insetti) a insetticidi organofosfati. L'acetilcolina è molecola derivata dall'unione della colina<sup>6</sup> con l'acido acetico, in seguito all'azione di un enzima detto esterasi. Questa molecola interviene come mediatore chimico della trasmissione degli impulsi nervosi (detta in tal caso trasmissione colinergica), in molteplici punti del sistema nervoso centrale e periferico. Questi insetticidi sono derivati dall'acido fosforico e sono ampiamente utilizzati in agricoltura e in settori zootecnici. La loro tossicità si manifesta mediante l'inibizione

<sup>6</sup>La *colina* è una sostanza organica classificata come nutriente essenziale. È un costituente dei lipidi che compongono la membrana cellulare e del neurotrasmettitore acetilcolina.

dell'acetilcolinesterasi degli insetti provocando una stimolazione continua e talvolta eccessiva dei recettori con conseguenze rilevanti sulle funzioni motorie.

Recenti studi hanno messo in evidenza che i sistemi enzimatici, oltre ad essere utilizzati come biomarcatori di pesticidi, possono trovare impiego anche nel monitoraggio di certi metalli pesanti. Ad esempio, studi condotti da gasteropodi marini[8] hanno dimostrato che gli enzimi colinesterasici vengono inibiti ad elevate concentrazioni di cadmio e rame portando a squilibri mentali, paralisi e anomalie comportamentali.



## Capitolo 3

# Informatica e biologia

*“La **bioinformatica** è lo studio del contenuto informativo e del flusso informativo in sistemi e processi biologici”.*

*Hwa Lim*

Esiste un forte legame tra la biologia e l'informatica.

Da una parte, i computer hanno caratteristiche simili a quelle delle cellule, le macchine fondamentali in natura. Come il software, le cellule influenzano, determinano, causano, programmano e modellano altri comportamenti.

Tutti i computer hanno una struttura e funzioni di base simili, ma possono eseguire una vasta gamma di attività. Analogamente, tutte le cellule hanno una struttura di base simile, tuttavia possono sopravvivere in ambienti radicalmente differenti o svolgere un'ampia varietà di funzioni. Di conseguenza, le astrazioni, gli strumenti e i metodi utilizzati per specificare e studiare i sistemi informatici dovrebbero permettere di gettare nuova luce sulle nostre conoscenze relative ai sistemi biomolecolari[1].

Tutto questo ebbe inizio nei primi anni '50 grazie a Von Neumann[?] che, per capire le origini della vita, pensò alla possibilità di costruire una “macchina” (automa cellulare) che contenesse una completa descrizione di se stessa e capace di autoriproduzione e assemblaggio secondo un

meccanismo che in seguito risultò essere lo stesso di quello biologico scoperto da James Watson e Francis Crick<sup>1</sup> (premi Nobel per la medicina nel 1962).

Da allora, gli organismi biologici e le loro funzioni sono modellati attraverso i linguaggi artificiali della logica e della matematica (compresi i processi cognitivi nel settore dell'Intelligenza Artificiale) e spesso implementati in computer che permettono la simulazione di comportamenti complessi. Inoltre la biologia ha bisogno delle basi di dati<sup>2</sup>, di algoritmi e sistemi esperti per cercare di estrarre dati significativi e fare predizioni da insiemi spuri di dati prodotti da esperimenti[2]. Le predizioni possono essere fatte mediante un ragionamento induttivo, partendo dai dati memorizzati e seguendo delle regole che descrivono il sistema biologico in esame.

## 3.1 Programmazione Logica

La logica del primo ordine è stata presa come base per costruire in modo semplice e intuitivo rappresentazioni del mondo e usare un processo di inferenza per affermare la verità o la falsità di affermazioni relative al mondo rappresentato.

Nella rappresentazione della conoscenza un *dominio* è una sezione del mondo a proposito della quale vogliamo sperimentare della conoscenza (le

---

<sup>1</sup>Studiarono insieme la struttura del DNA scoprendo che è composto da una doppia elica e che alcune parti sono sempre collegate in coppie. A seguito di queste scoperte riuscirono a teorizzare un modo in cui il DNA duplica se stesso e il processo mediante il quale vengono ereditati i geni.

<sup>2</sup>Storicamente, la prima raccolta di dati biologici risale al 1960 grazie a *Margaret Dayhoff*, studiosa di evoluzione molecolare presso la National Biology Research Foundation. Le prime due banche dati di sequenze di acidi nucleici furono: la banca dati Europea (EMBL Datalibrary) e la banca dati Americana (GenBank).

parentele, gli insiemi matematici, . . . , un sistema biologico).

La logica del primo ordine, o almeno sostanziali sottoinsiemi di essa, è usata come linguaggio di programmazione: programmazione logica.

### 3.1.1 Prolog

La prima e tuttora più diffusa implementazione della programmazione logica è il linguaggio Prolog.

Il Prolog (PROgramming in LOGic) fu progettato ed implementato a Marsiglia da Colmerauer e Roussel nel 1972. È basato sul principio di risoluzione di Robinson del 1965 e sull'interpretazione data da Kowalski della logica dichiarativa come insieme di istruzioni procedurali per un calcolatore.

La sintassi del Prolog è formata da una importante classe di formule della logica del primo ordine chiamate *clausole*. Vediamo, con l'aiuto di un esempio (Figura. 3.1), le diverse tipologie di clausole<sup>3</sup>:

**I fatti** La clausola più semplice è un *fatto*. I fatti sono un mezzo per asserire che esiste una relazione tra oggetti nel dominio che stiamo rappresentando oppure per esprimere una proprietà dell'oggetto. Un insieme finito di fatti costituisce un *programma*. Osservando la Figura. 3.1 vediamo che

*maschio(michele).*  
*genitore(michele, davide).*

il primo fatto indica che Michele appartiene al genere maschile (esprime una proprietà di Michele) mentre il secondo fatto ci dice

---

<sup>3</sup>Una trattazione più formale dei linguaggi logici si trova in [18].

```
(1) genitore(michele,davide).  
    genitore(sabrina,davide).  
    genitore(sabrina, lindia).  
    genitore(davide, anna).  
    genitore(davide, teresa).  
    genitore(teresa, giorgio).  
(2) figlio(Y,X) :- genitore(X,Y).  
(3) nonno(X,Z) :- genitore(X,Y),  
                  genitore(Y,Z).  
(4) hafigli(X) :- parent(X,Y).  
(5) antenato(X,Z) :- genitore(X,Z).  
    antenato(X,Z) :- genitore(X,Y),  
                      antenato(Y,Z).  
(6) maschio(michele).  
    maschio(davide).  
    maschio(giorgio).  
(7) femmina(sabrina).  
    femmina(linda).  
    femmina(anna).  
    femmina(teresa).
```

Figura 3.1: Descrizione di una famiglia tramite un programma scritto in Prolog

che Michele è genitore di Davide. *maschio* indica una proprietà che un oggetto del dominio può avere, *genitore* è il nome della relazione mentre *michele* e *davide* sono i suoi argomenti. Da questo insieme di fatti si può vedere che Teresa non è figlia di Michele ma è figlia di Davide.

**Le regole** Una regola consente di definire nuove relazioni sfruttando relazioni già esistenti. Come si può osservare dalla Figura. 3.1 le regole hanno la forma

$$\underbrace{\text{figlio}(Y, X)}_{\text{testa}} \text{ :- } \underbrace{\text{genitore}(X, Y)}_{\text{corpo}}.$$

Il corpo della regola rappresenta la condizione che deve essere vera affinché la conclusione della regola, la testa, sia vera. La regola presentata sopra si può leggere in questo modo: “*Per tutti gli  $X, Y$  se  $X$  è genitore di  $Y$  allora  $Y$  è figlio di  $X$* ”. Nel corpo di una regola possono esserci più condizioni separate da una virgola. Affinché la conclusione sia vera, devono essere vere tutte le condizioni presenti nel corpo. Sempre in relazione al nostro esempio sulla famiglia: “*una persona  $X$  è nonno di una persona  $Z$  se:  $X$  ha un figlio,  $Y$ , e quest’ultimo è padre di  $Z$* ”.

Una regola è *ricorsiva* se la relazione da essa espressa viene descritta in termini di se stessa ovvero se nel corpo della regola è presente il nome della relazione che la regola sta definendo. La regola che definisce la relazione *antenato*, indicata con (5) nel nostro esempio, è di questo tipo.

**Le richieste** Tramite le richieste si riesce ad estrarre informazione da un programma logico. Una richiesta serve per sapere se esiste una certa relazione tra oggetti. Ad esempio

?- genitore(michele, davide).

chiede se esiste la relazione *genitore* tra Michele e Davide. Dati i fatti del programma (a) nella Figura. 3.1, la risposta a questa richiesta sarà “*si*”. Nel caso in cui la richiesta contenga una variabile come ad esempio

?- genitore(X, teresa).

la risposta del Prolog non sarà “*si*” o “*no*” ma assegnerà ad X un valore che renda vera la richiesta se esiste, altrimenti risponderà “*no*”. In relazione al nostro esempio, X avrà valore “davide”. Una richiesta può avere più risposte corrette che si possono ottenere sequenzialmente; quando non ci saranno più soluzioni disponibili il Prolog risponderà “*no*”.

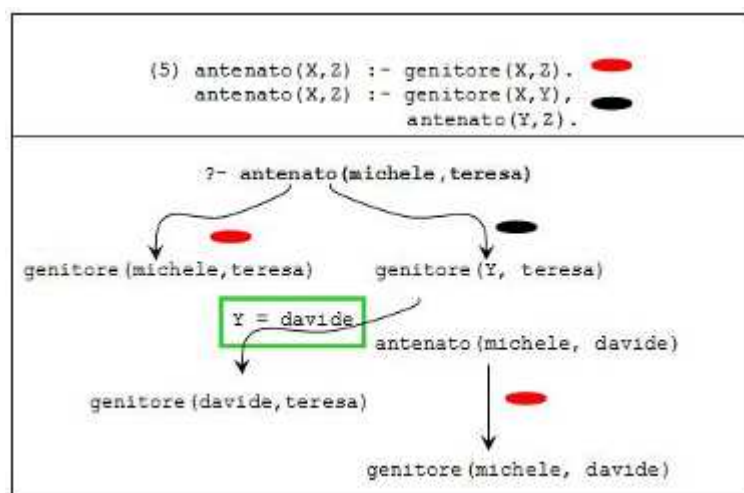


Figura 3.2: Albero di ricerca per la richiesta `?- antenato(michele,teresa)`.

Rispondere ad una richiesta in relazione a un programma significa determinare se la richiesta è **conseguenza logica** del programma. Le conseguenze logiche si ottengono dall'applicazione delle regole di deduzione. La strategia utilizzata dal Prolog per il calcolo di una soluzione si chiama **SLD**<sup>4</sup>.

Lo spazio di ricerca delle soluzioni si può vedere come un albero (Figura. 3.2) nel quale alla radice troviamo le richieste e ogni ramo rappresenta il corpo di una regola la cui testa è uguale alla richiesta

<sup>4</sup>È un caso particolare del principio di risoluzione utilizzato per automatizzare la dimostrazione di teoremi introdotto da Robinson nel 1965[19]

effettuata. Tramite una lista di sostituzioni il Prolog cerca di ottenere la richiesta partendo dal programma. Il Prolog utilizza la ricerca in profondità, da sinistra a destra, tornando indietro alla ramificazione appena precedente se arriva a un punto morto. La ricerca in profondità rende il Prolog incompleto in quanto potrebbe procedere lungo un cammino infinito non trovando soluzioni quando in realtà queste esistono ma su un cammino diverso da quello intrapreso.

Vediamo in Figura. 3.2 l'albero per la richiesta

?- *antenato(michele,teresa)*. Come si può osservare, i due rami dell'albero sono i corpi delle due regole che hanno come testa il nome "*antenato*". Vediamo che la prima regola fallisce (ramo di sinistra) in quanto non esiste il fatto *genitore(michele, teresa)* nel nostro programma. Il corpo della seconda regola ha due condizioni e nella ricerca della soluzione le condizioni vengono analizzate secondo l'ordine in cui sono scritte: prima viene cercato un valore che renda vera la condizione *genitore(Y, teresa)* e dai fatti del programma si trova che solo per  $Y = \text{davide}$  la condizione viene soddisfatta. Nella condizione successiva andremo a sostituire il valore appena trovato, così la nuova condizione da soddisfare sarà *antenato(michele, davide)*. Quest'ultima viene soddisfatta dal fatto *genitore(michele, davide)*. Se questo fatto non fosse stato presente nel programma, il Prolog avrebbe risposto "*no*" alla richiesta invece di " $Y = \text{davide}$ ".

L'ordine con il quale il Prolog restituisce le possibili soluzioni dipende dall'ordine delle regole all'interno del programma e dall'ordine delle richieste. Un ordine sbagliato nelle regole del programma può rendere molto inefficiente l'esecuzione del programma se non addirittura renderne impossibile la terminazione. Le regole ricorsive possono

portare facilmente ad una situazione del genere se non sono definite correttamente. Sapendo che le regole vengono applicate nell'ordine in cui sono trovate, bisogna definire sempre la regola che descrive il caso base della ricorsione e successivamente la regola ricorsiva.

### 3.1.2 Applicazioni del Prolog

Potremmo fare una lista infinita delle applicazioni in cui il Prolog viene utilizzato ma per questioni di spazio e praticità ne elenchiamo solo alcune:

- *Giochi*: il problema delle N regine, scacchi, otello, dama, ...
- *Dimostratori di teoremi*: il programma SAM (Semi-Automated Mathematics) è stato il primo a dimostrare un lemma della teoria dei reticoli. Il dimostratore di teoremi di Boyer-Moore (1979) è stato usato ed esteso per molti anni, usato anche da Natarajan Shankar per dare la prima dimostrazione formale e completamente rigorosa del teorema di incompletezza di Godel (1986). I dimostratori di teoremi possono essere applicati a problemi legati alla verifica e alla sintesi sia di hardware che di software, poiché di entrambi i domini si possono dare assiomatizzazioni corrette.
- *Pianificazione*: i sistemi di pianificazione e scheduling sono stati usati largamente nella pianificazione di missioni spaziali ed anche nella costruzione di navicelle spaziali (Voyager, UOSAT-II e ERS-1)
- *Analisi del linguaggio naturale*: l'analisi del linguaggio naturale si basa sulla definizione di grammatiche le quali descrivono le frasi ammesse e quelle sbagliate. Si può usare il Prolog per esprimere le grammatiche facendo corrispondere una regola del Prolog a ciascuna



produzione della grammatica. Una delle prime applicazioni dell'analisi del linguaggio naturale fu il programma “*ELIZA*” ideato da Joseph Weizenbaum[20]

- *Logica giuridica*: le prime ricerche in materia di Intelligenza Artificiale applicate al diritto risalgono agli anni '70[11] dove si cerca di automatizzare il ragionamento giuridico mediante sistemi basati su regole[12].
- *Descrizione di sistemi biologici*: il programma di apprendimento *GOLEM* descrive le proprietà dei legami tra enzimi e substrati[9] mentre in [15] si spiega con l'inferenza il ruolo degli enzimi all'interno di un ciclo metabolico. BIOCHAM è un ambiente di sviluppo<sup>5</sup> scritto in Prolog e Java per la modellazione di sistemi biochimici, simulazioni e controllo temporale di proprietà. Altri esempi di applicazione della logica nel campo della biologia si trovano in [13], [17].
- ...

---

<sup>5</sup>È possibile scaricarlo gratuitamente dal sito <http://contraintes.inria.fr/BIOCHAM/>.

# Capitolo 4

## Descrizione di “*EnzimiLab*”

L’idea dell’implementazione di questa applicazione nasce dalla collaborazione con il team di ricerca del Dipartimento di Biologia dell’Università di Pisa, guidato dal Prof. Umberto Mura<sup>1</sup>, all’interno del progetto di ricerca “*Studio di metodologie innovative per la valutazione dell’esposizione dei lavoratori ad agenti biologici e chimici derivanti da attività produttive correlate allo smaltimento dei rifiuti*”.

Lo scopo di questo progetto è l’identificazione di enzimi che, a seguito della loro suscettibilità all’azione di inquinanti e composti tossici vari, possano essere utilizzati come biomarcatori nella realizzazione di un sensore multi enzimatico.

Sono stati presi in esame enzimi che svolgono ruoli diversi nel metabolismo di molti essere viventi determinando l’effetto di diverse concentrazioni di potenziali agenti inquinanti (chiamati in seguito anche *effettori*), sia organici che inorganici, sulla loro attività.

---

<sup>1</sup>Il gruppo di ricerca è composto dal Prof. Umberto Mura, dalla Prof. Antonella del Corso, dal Dr. Mario Cappiello, dalla Dr.ssa Roberta Moschini, dalla Dr.ssa Michela Gianecchini e dalla Dr.ssa Elisa Di Bugno.

Gli esperimenti sono stati eseguiti utilizzando una concentrazione degli inquinanti al limite (Tabella. 4.1) e a *10 volte*<sup>2</sup> la concentrazione limite di legge (10X).

Valori delle concentrazioni limite per legge		
Metallo	mg/l	[μM]
Argento	1	37
Bario	20	146,5
Boro	4	370
Cadmio	0,1	0,89
Cobalto	2	34
Cromo	2	38,5
Ferro	2	35,8
Fluoro	1,5	79
Magnesio	50	2057
Manganese	2	36,4
Mercurio	0,005	0,025
Molibdeno	1	10,4
Nichel	2	34,1
Piombo	0,5	2,4
Rame	1	15,7
Selenio	0,03	0,38
Stagno	10	84,2
Vanadio	0,1	19,6
Zinco	2	30,6

Figura 4.1: Valori delle concentrazioni limite per legge dei metalli.

Questi esperimenti sono stati eseguiti secondo due modalità:

**Diretta:** si misura direttamente l'azione dell'effettore sul potere catalitico dell'enzima

---

<sup>2</sup>Vengono utilizzare due concentrazioni diverse (limite di legge e 10 volte superiore il limite di legge) perché non sempre per concentrazioni al limite di legge si hanno degli effetti negativi sull'attività catalitica degli enzimi.

**Preincubazione:** l'azione dell'effettore viene valutata dopo una preincubazione dell'effettore con l'enzima per 30 minuti con lo scopo di analizzare eventuali danni alla conformazione strutturale dell'enzima

Per ogni coppia (enzima,effettore) sono state effettuate più prove formando una quantità di dati piuttosto grande. L'attività catalitica di ogni enzima può essere rappresentata graficamente come un cerchio (Figura. 4.2), dove per ogni effettore viene tracciato un raggio: il centro del cerchio rappresenta un'attività nulla dell'enzima mentre l'intersezione con la circonferenza indica che l'enzima è totalmente attivo anche in presenza dell'effettore associato al raggio. Per ogni enzima avremo quattro cerchi che rappresentano il suo livello di attività in presenza dei diversi effettori in base al tipo di esperimento (diretto o con preincubazione) e al livello di concentrazione dell'effettore (limite di legge o 10 volte superiore).

Tutti questi dati si possono utilizzare per cercare di indurre le possibili combinazioni di sostanze chimiche inquinanti presenti in un refluo avendo come dati di partenza solo le attività degli enzimi rispetto al refluo analizzato. Vista la grande quantità di dati da trattare diventa necessario o perlomeno auspicabile un aiuto di tipo informatico.

L'applicazione sviluppata per questa tesi ha come scopo principale quello di proporre una serie di possibili combinazioni di effettori che potrebbero comporre un refluo ricevendo in input una serie di enzimi (di cui abbiamo i dati degli esperimenti) con i relativi livelli di attività nel refluo e basandosi sui dati degli esperimenti effettuati dai biologi in laboratorio.

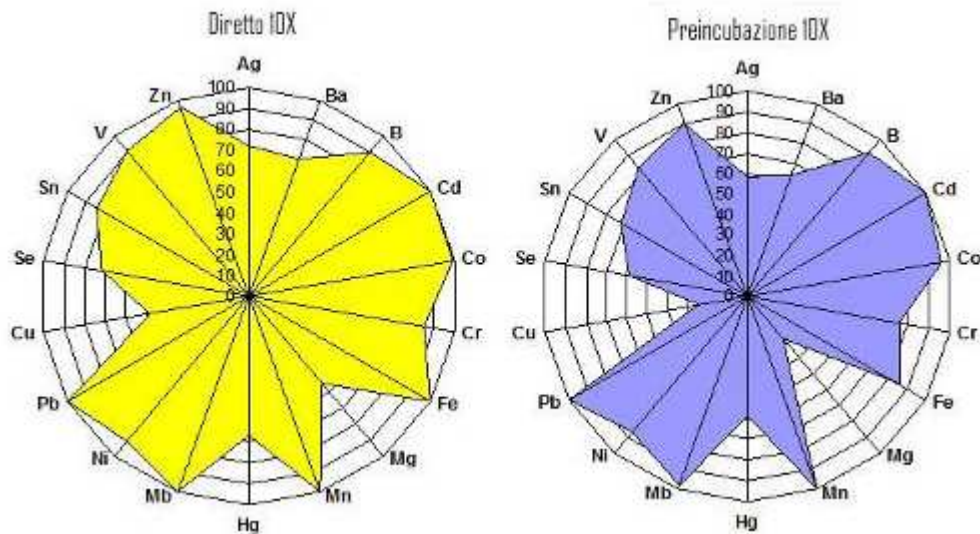


Figura 4.2: Questi due cerchi rappresentano i livelli di attività dell'enzima *tirosinasi* rispetto ad effettori metallici ad una concentrazione 10 volte superiore il limite di legge. Il cerchio sulla sinistra riporta i risultati degli esperimenti di tipo *diretto* mentre quello destra i risultati degli esperimenti con *preincubazione*.

## 4.1 Il nucleo dell'applicazione

Il modello logico per l'analisi della composizione del refluo è stato implementato utilizzando il Prolog (in particolare l'implementazione SICStus Prolog 3.11.2[24]).

Questo modello, per convenienza, si può vedere suddiviso in tre moduli:

- I dati del modello (enzimi, effettori, esperimenti)
- Le osservazioni sul refluo di input (setup refluo)
- La teoria del refluo che lega le osservazioni di input alla base di conoscenza per indurre la composizione del refluo (teoria refluo)

### 4.1.1 Struttura del modello logico

Prima di spiegare l'algoritmo che calcola la composizione del refluo di input, vediamo quali sono le componenti del modello.

#### I dati del modello

L'insieme degli enzimi e degli effettori sono rappresentati rispettivamente dai predicati *enzima/1*<sup>3</sup> e *effettore/1*. Entrambi i predicati hanno un solo argomento che rappresenta il nome, rispettivamente, dell'enzima e dell'effettore.

Queste due entità del dominio sono suddivise in due file: "enzimi.pl" e "effettori.pl".

Il predicato *attivit \_enzima/4* raggruppa una serie di esperimenti relativi all'enzima *enz\_i*, l'effettore *eff\_j* i quali portano al calcolo dell'attività media dell'enzima in analisi con un certo margine di errore, la deviazione standard.

Il predicato non precisa la concentrazione dell'effettore e nemmeno la tipologia dell'esperimento (diretto o con preincubazione) in quanto queste clausole vengono suddivise in file separati chiamati: "diretti1x.pl", "diretti10x.pl", "incubazione1x.pl" e "incubazione10x.pl". Questa scelta è stata possibile in quanto il refluo di input sarà indirizzato *unicamente* ad una di queste quattro possibilità quindi ho preferito rendere il predicato più semplice non mettendo queste indicazioni come argomenti e nemmeno creare quattro predicati diversi per indicare lo stesso significato nel modello.

---

<sup>3</sup>In Prolog la descrizione di un predicato ha la forma *nome\_predicato/numero\_argomenti*.

### Le osservazioni

Nel modello, un refluo viene rappresentato con una sequenza di coppie  
(enzima, livello della sua attività nel refluo in analisi).

La rappresentazione grafica data in Figura. 4.3 fornisce subito un'idea della qualità del refluo: più l'area viola si avvicina alla circonferenza e più puro (meno inquinato) è il refluo.

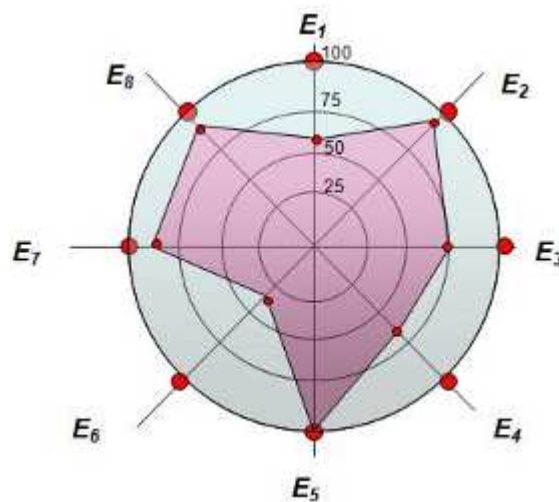


Figura 4.3: Dato un refluo, si seleziona un insieme di enzimi e si misura il loro livello di attività nel refluo. Ogni raggio nel cerchio rappresenta il livello di attività dell'enzima, più il raggio è lontano dalla circonferenza meno l'enzima è attivo mentre se il raggio tocca la circonferenza significa che il refluo non ha alcun effetto su tale enzima.

Il predicato che elabora l'input ed inserisce le osservazioni a complemento dei dati degli esperimenti è *elabora\_coppia/2*:

```
elabora_coppia(Enzima,Perc_attiva) :-
    enzima(Enzima),
    asserta(attivita_enzima_reflue(Enzima,Percentuale_attiva)).
```

Questo predicato si trova nel file “setup-refluo.pl” ed ha come argomenti il nome dell'enzima (una costante) e il livello della sua attività nel reflu analizzato. Tale attività varia nell'intervallo  $[0.0, +\infty[$  dove il valore minimo dell'intervallo segnala che l'enzima è totalmente disattivato dal reflu, un valore pari a 100.0 indica invece che l'attività enzimatica non è intaccata mentre un valore superiore a 100.0 rivela che il potere catalitico dell'enzima è maggiore rispetto alla normalità. Se l'enzima appartiene alla base di conoscenza (i dati del modello) allora inserisce l'osservazione mediante il predicato dinamico *attivita\_enzima\_refluo*.

### La teoria

La teoria si occupa del calcolo della composizione del reflu sfruttando la base di conoscenza precedentemente introdotta.

Dato un enzima, l'insieme dei possibili effettori che compongono il reflu e che influiscono sul suo potere catalitico si possono suddividere in tre gruppi: gli effettori che non hanno alcun effetto sull'attività dell'enzima (*effettori innocui*), quelli che da soli riescono ad avere l'effetto osservato (*effettori singoli*) e le possibili combinazioni di effettori che sommando insieme i loro effetti riescono a produrre l'effetto osservato ovvero a portare l'attività dell'enzima a quella rilevata nel reflu (*combinazioni additive di effettori*). I predicati che compongono questa parte li vedremo in seguito, dopo aver dato un'idea intuitiva di come lavora l'algoritmo.

#### 4.1.2 L'algoritmo per il calcolo della composizione del reflu

L'algoritmo procede per passi scremando ogni volta le informazioni raccolte precedentemente fino ad arrivare ad una soluzione che rispetta certe



condizioni, che saranno illustrate in seguito, in relazione a *tutti* gli enzimi in esame. I passi principali in cui si suddivide l'algoritmo sono:

1. Selezione di un enzima (quello con l'attività massima nel refluo) e calcolo della lista degli effettori che potrebbero appartenere al refluo (*effettori candidati*).
2. Dalla lista degli effettori candidati si estraggono quelli innocui (*effettori innocui*).
3. La lista degli effettori innocui viene testata sul resto degli enzimi ottenendo così la lista degli effettori innocui per *tutti* gli enzimi analizzati. Gli effettori che non sono innocui per tutti vengono rimessi nella lista degli effettori candidati.
4. Anche la lista degli effettori candidati viene testata sul resto degli enzimi ottenendo così la lista degli effettori candidati per *tutti* gli enzimi.
5. Dagli effettori candidati si estraggono altre due liste: quella degli effettori singoli e quella delle combinazioni additive degli effettori (in relazione al primo enzima selezionato).
6. Una volta terminato il calcolo delle due liste, si scorrono gli enzimi rimanenti e da ogni lista si eliminano eventualmente gli effettori (o una loro combinazione) che per l'enzima attualmente analizzato non rispettano determinate condizioni (spiegate in seguito).
7. L'output del programma consiste delle tre liste create ai passi precedenti, ovvero: lista degli effettori innocui, lista degli effettori singoli e lista delle combinazioni additive di effettori.

### Selezione degli effettori candidati

Questo è lo pseudocodice che seleziona i possibili effettori:

```
(1) input [(enzima1, attivita1), ..., (enzima_n, attivita_n)];
(2) enzima_analizzato = enzima_massima_attivita(input);
(3) attivita_analizzata = attivita_massima(input);
(4) output [effettori_candidati];

per ogni predicato('attivita_enzima(enzima, effettore, media, dev_std)')
{
  (5) se enzima == enzima_analizzato allora
  {
    (6) valore_massimo = media + dev_std;
    (7) se valore_massimo >= attivita_analizzata
      (8) effettori_candidati = add(effettori_candidati, effettore);
  }
}
```

Un effettore può far parte della composizione del refluo se riduce l'attività dell'enzima al più quanto quest'ultima viene ridotta dal refluo, perciò *valore\_massimo*(6), che rappresenta il massimo valore che l'attività dell'enzima può raggiungere in presenza di tale effettore, deve essere maggiore o uguale al valore dell'attività dell'enzima nel refluo (*attivita\_analizzata*(3)).

Come si può osservare nelle Figure.4.4 e 4.5 , vengono presi tutti gli effettori il cui quadratino azzurro si trova al di sopra del puntino rosso. Gli altri effettori vengono scartati<sup>4</sup> in quanto riducono maggiormente l'attività dell'enzima quindi l'enzima, in loro presenza dovrebbe avere un valore minore di quello che invece è stato osservato nel refluo.

---

<sup>4</sup>Gli effettori scartati sono: Argento (Ag), Bario (Ba), Magnesio (Mg), Mercurio (Hg), Rame (Cu) e Selenio (Se).

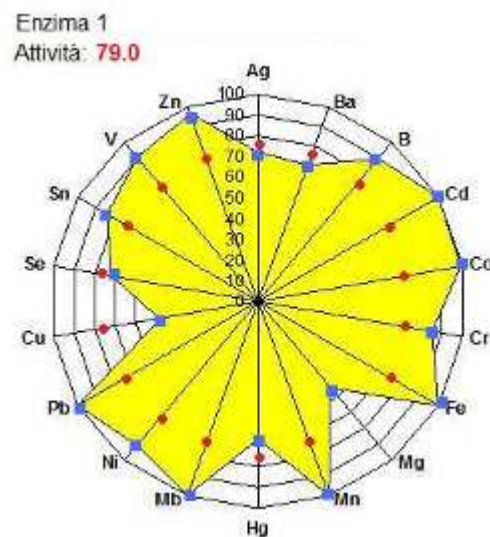


Figura 4.4: L'enzima in input ha un livello di attività pari a 79.0 e sul grafico viene evidenziata con un pallino rosso. Con il quadratino azzurro viene evidenziato il livello di attività dell'enzima in presenza del corrispettivo effetto.

### Selezione degli effettori innocui

Una volta calcolata la lista degli effettori che potrebbero comporre il reflu, estraggo quegli effettori la cui presenza non incide negativamente sull'attività dell'enzima. Alla fine di questa fase avremo due liste distinte: quella degli effettori innocui e quella dei restanti effettori candidati che sarà elaborata dai passi successivi della computazione.

Di seguito vediamo in pseudocodice le operazioni effettuate in questa fase:

```
(1) input enzima_analizzato, effettori_candidati
(2) output [effettori_innocui], [effettori_candidati];

per ogni predicato('attivita_enzima(enzima, effettore, media, dev_std)')
{
    (3) se enzima == enzima_analizzato allora
```

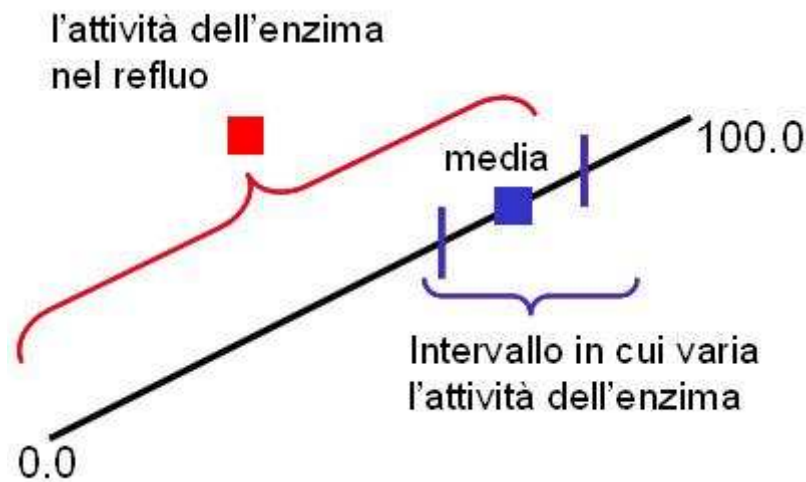


Figura 4.5: Se l'attività dell'enzima nel refluo è inferiore o uguale alla massima attività dell'enzima in presenza dell'effettore analizzato allora quest'ultimo è candidato.

```

{
  (4) valore_minimo = media - dev_std;
  (5) se valore_minimo >= 100.0
  {
    (6) effettori_innocui = add(effettori_innocui, effettore);
    (7) effettori_candidati = remove(effettori_candidati, effettore);
  }
}

```

Un effettore è innocuo per l'enzima se la sua presenza non diminuisce il suo potere catalico. Il valore *valore\_minimo*(4) rappresenta la minima attività che l'enzima può avere in presenza di un determinato effettore; se tale valore è superiore o pari a 100.0 allora l'effettore viene classificato come *innocuo* in quanto la sua presenza/assenza nel refluo non incide in negativo sull'enzima in esame e quindi verrà aggiunto alla *lista degli effettori innocui* e tolto dall'insieme degli effettori candidati. Successivamente gli effettori

appartenenti a questa lista vengono testati sul resto degli enzimi analizzati ottenendo come risultato finale gli effettori innocui rispetto a *tutti* gli enzimi. Gli effettori che vengono scartati tornano a concorrere per le combinazioni additive. Facendo sempre riferimento alla Figura. 4.4, sono innocui<sup>5</sup> gli effettori il cui quadratino azzurro si trova sulla circonferenza.

### Selezione degli effettori singoli

Gli effettori singoli sono quelli in grado di diminuire il potere catalitico dell'enzima della stessa quantità rilevata dal reflu. Più precisamente la condizione che viene valutata è alla *riga(6)* dello pseudocodice seguente:

```
(1) input enzima_analizzato, attivita_analizzata, effettori_candidati;
(2) output [effettori_singoli];

per ogni predicato('attivita_enzima(enzima, effettore, media, dev_std)')
{
  (3) se enzima == enzima_analizzato allora
  {
    (4) valore_massimo = media + dev_std;
    (5) valore_minimo = media - dev_std;
    (6) se (valore_massimo <= attivita_analizzata &&
          valore_minimo >= attivita_analizzata) allora
      (7) effettori_singoli = add(effettori_singoli, effettore);
  }
}
```

In questa fase, per ogni effettore viene calcolato l'intervallo della variazione della velocità dell'enzima in sua presenza. Se l'attività dell'enzima all'interno del reflu analizzato cade in questo intervallo significa che, per questo enzima, potrebbe esserci anche solo questo effettore a comporre il reflu.

---

<sup>5</sup>Gli effettori innocui sono: Cadmio (Cd), Cobalto (Co), Ferro (Fe), Manganese (Mn), Molibdeno (Mb) e Piombo (Pb).

Nel caso in cui un esperimento abbia deviazione standard nulla, l'effettore ad esso relativo sarà difficilmente selezionato come singolo in quanto dovrebbe succedere che l'enzima nel refluò abbia *esattamente* il valore rilevato al momento dell'esperimento.

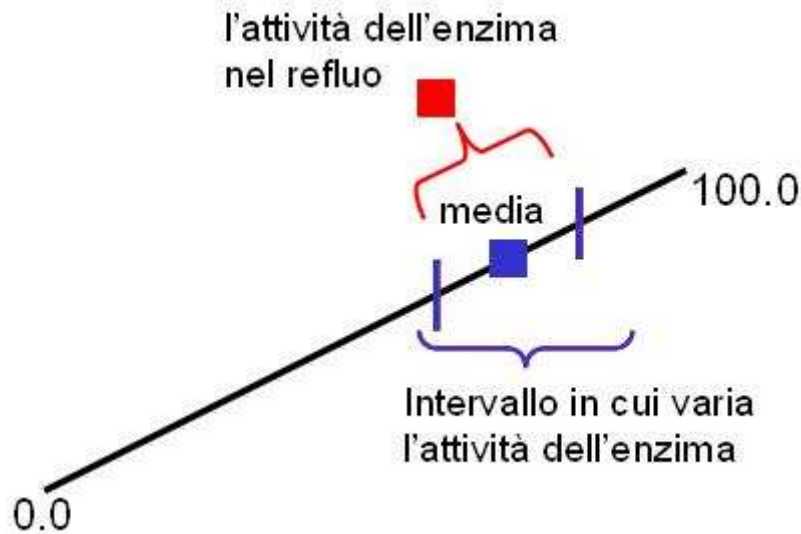


Figura 4.6: Se l'attività dell'enzima nel refluò cade nell'intervallo in cui l'attività dell'enzima varia in presenza di un effettore allora quest'ultimo potrebbe essere l'unica sostanza presente nel refluò.

Facendo riferimento alla Figura. 4.6, un effettore viene selezionato come *singolo* quando il puntino rosso cade nell'intervallo azzurro e nel caso di deviazione standard nulla, i due punti, rosso e azzurro, dovrebbero coincidere. Osservando la Figura. 4.4 gli effettori singoli potrebbero essere il Cromo (Cr) e il Boro (Bo).

### Selezione delle combinazioni additive di effettori

L'elaborazione delle possibili combinazioni di effettori che potrebbero formare il refluò è la parte più interessante e difficile in quanto, parlando

appunto di *combinazioni* che per loro natura sono *esponenziali* si è dovuta cercare una soluzione meno banale che della semplice generazione di tutte le combinazioni degli effettori presenti nella lista dei candidati<sup>6</sup>.

Sempre sottoforma di pseudocodice viene mostrata di seguito la logica con cui vengono create le combinazioni e i criteri di selezione delle combinazioni additive che poi formeranno il risultato finale.

```
(1) input enzima_analizzato, attivita_analizzata, [effettori_candidati];
(2) ammanco_enzima = 100.0 - attivita_analizzata;
(3) numero_effettori_candidati = lunghezza(effettori_candidati);
(4) k = 2; /* lunghezza iniziale della combinazione */
(5) combinazioni_da_estendere = [];
(6) comb_k = crea_combinazioni_ordinate(effettori_candidati, k, combinazioni_candidate);
(7) output [combinazioni_additive_effettori];

(8) per ogni combinazioni_candidate('comb')
{
    somma_massima = ( per ogni effettore della combinazione calcolo l'ammanco
                      (100.0 - (media + deviazione standard)) e li sommo
                      );

    se somma_massima <= ammanco_enzima allora
    {
        somma_minima = ( per ogni effettore della combinazione calcolo l'ammanco
                          (100.0 - (media - deviazione standard)) e li sommo
                          );

        se somma_minima >= ammanco_enzima allora
            (8') combinazioni_additive_effettori = add(combinazioni_additive_effettori, comb);

            (8'') combinazioni_da_estendere = add(combinazioni_da_estendere, comb);
    }
}
```

---

<sup>6</sup>Generando tutte le possibili combinazioni di 12 elementi, l'interprete Prolog terminava con “*memory overflow*”.

```

(9) per ogni combinazioni_da_estendere('comb')
{
    se lunghezza(comb) < numero_effettori_candidati allora
    {
        ultimo = get_last_element(comb); /* la combinazione è ordinata! */
        posizione = index(ultimo, effettori_candidati);
        se posizione < numero_effettori_candidati allora
        {
            elementi_successivi = get_elementi_successivi(posizione, effettori_candidati);
            per ogni elementi_successivi(nuovo_elemento)
                add(comb,nuovo_elemento);
        }
        else elimina_combinazione(combinazioni_da_estendere, comb);
    }
}

se combinazioni_da_estendere != [] allora
{
    combinazioni_candidate = combinazioni_da_estendere;
    goto (8);
}

```

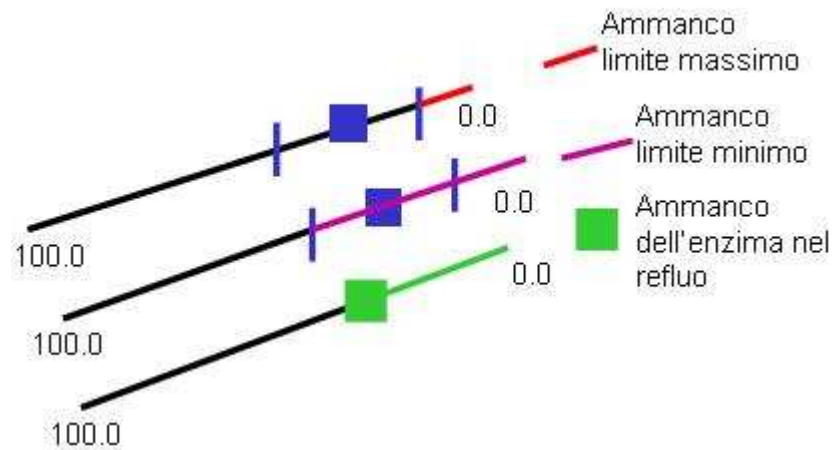


Figura 4.7: Definizione di ammanco dell'attività enzimatica e degli ammanchi dell'attività dell'enzima relativamente ad un effettore.



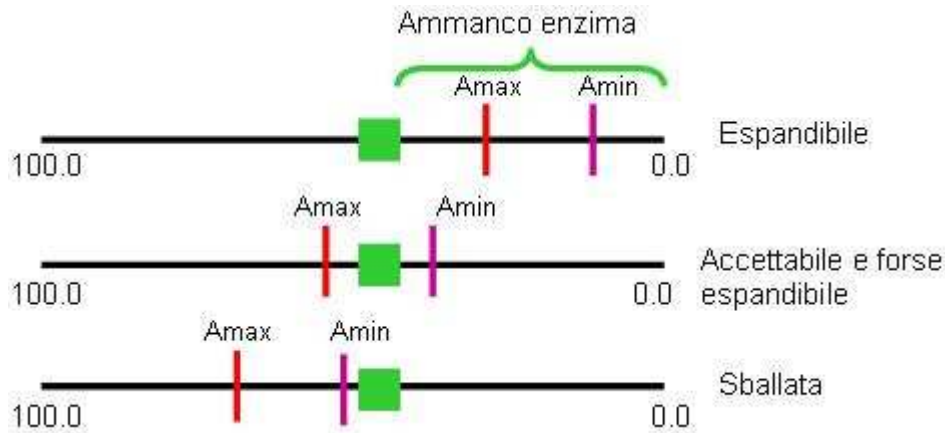


Figura 4.8: La finestra rappresenta la quantità di ammanco coperta da un insieme di effettori. Finché la finestra è sotto l'attività dell'enzima nel refluo, occorre aggiungere effettori, mentre quando l'intervallo comprende l'attività dell'enzima la combinazione di effettori corrispondente può essere considerata esaustiva anche se ancora espandibile. Quando la finestra supera l'attività dell'enzima la combinazione di effettori ad essa relativa viene scartata.

Le combinazioni additive vengono create in relazione all'ammanco dell'enzima (linea verde in Figura. 4.7), ovvero quanto manca all'attività dell'enzima per essere al massimo (100.0).

Un insieme di effettori può essere considerato una possibile composizione del refluo se la somma degli ammanchi dell'attività dell'enzima relativi a questi effettori è uguale all'ammanco che il refluo provoca all'enzima. Ovviamente, avendo la misura dell'errore (deviazione standard) basta che l'ammanco dell'enzima stia nell'intervallo  $[somma\_massima, somma\_minima]$  affinché l'insieme di tali effettori si possa considerare un'ipotesi accettabile dove per *somma\_massima* e *somma\_minima* si intende la somma degli ammanchi calcolati rispettivamente in relazione al massimo (linea rossa in Figura. 4.7) e al minimo (linea viola in Figura. 4.7) valore che l'attività dell'enzima può

avere in presenza di un certo effettore. Per semplicità, pensiamo a questo intervallo come ad una finestra scorrevole. Osservando la Figura. 4.8 vediamo che fino a quando la finestra si trova sotto il livello di attività dell'enzima, la combinazione di effettori viene estesa perchè la somma degli ammanchi da essi provocati non raggiunge l'ammanco attuale dell'enzima. Dal momento che l'attività dell'enzima cade nell'intervallo, la combinazione di effettori associata all'intervallo considerato può essere accettata anche se potrebbe essere ulteriormente espansa finchè l'aggiunta di un effettore non sposta così tanto la finestra da oltrepassare l'attività dell'enzima nel reflu. Se il valore di *somma\_massima* (barra viola in Figura. 4.8) supera l'attività dell'enzima (punto verde), la combinazione proposta relativa a questo valore verrà scartata.

Analizzando lo pseudocodice vediamo che partendo dall'insieme degli effettori candidati, vengono create tutte le combinazioni ordinate di lunghezza due (6). Successivamente, secondo i criteri spiegati precedentemente, le combinazioni sballate vengono ignorate mentre le altre saranno aggiunte alle combinazioni che faranno parte del risultato finale (8') se il loro valore cade nell'intervallo calcolato e in ogni caso le combinazioni che non sono scartate verranno aggiunte a quelle da espandere (8'').

## 4.2 Database

L'introduzione del database all'interno dell'applicazione è stata dettata da due esigenze principali:

- I dati storici del progetto in cui lo sviluppo di questa tesi si inserisce sono organizzati in diversi file .xls con formati eterogenei perché ovviamente sono stati inseriti nel lungo periodo e da persone diverse,

quindi risultano difficili da utilizzare da parte di un'applicazione informatica che nell'analizzare i dati ha bisogno di struttura.

- La memorizzazione dei dati in un database permette un accesso semplice e veloce, una gestione più omogenea e compatta e offre molti strumenti per effettuare ricerche sui dati. Ad esempio, se volessimo *“Tutti gli effettori che non fanno scendere l'attività enzimantica sotto una certa soglia”*, con l'organizzazione precedente avremmo dovuto aprire tutti i file .xls e andare a ricercare nelle tabelle tutti i valori mentre avendo a disposizione una base di dati otteniamo il risultato mediante una semplice query.

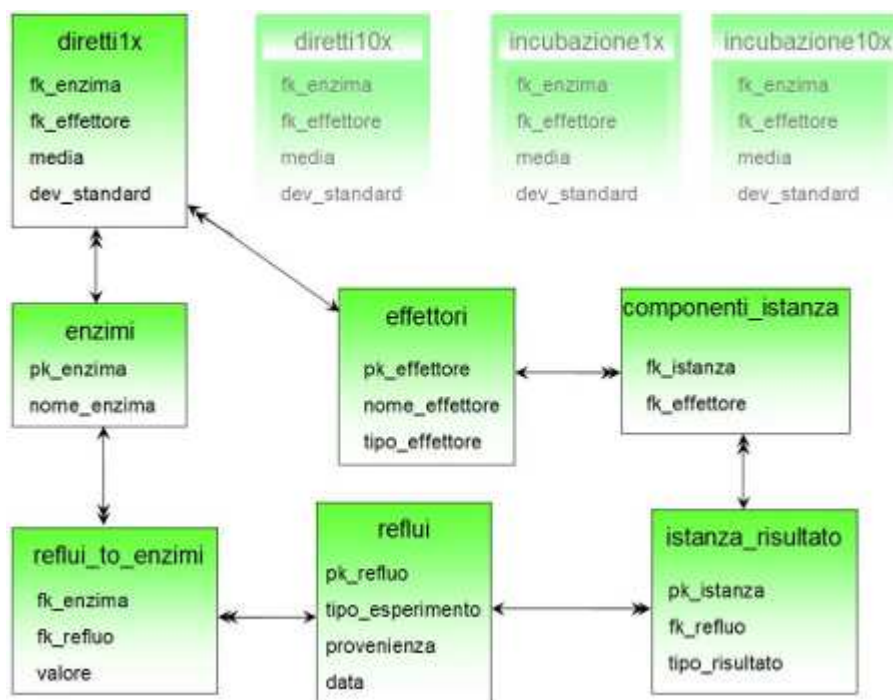


Figura 4.9: Schema del database di EnzimiLab.

Come database ho scelto *SQLite3*[27] essenzialmente perché è leggero, semplice e non necessita di un server per funzionare.

Le tabelle che compongono il database (Figura. 4.9) sono: *enzimi*, *effettori*, *diretti1x*, *diretti10x*, *incubazione1x*, *incubazione10x*, *reflui*, *reflui\_to\_enzimi*, *istanza\_risultato* e *componenti\_istanza*. Vediamo in dettaglio il ruolo e la struttura di ciascuna tabella.

**Enzimi:** la tabella *enzimi* memorizza il nome di tutti gli enzimi che sono stati presi come possibili biomarcatori. Ciascun enzima è descritto da due campi:

- *pk\_enzima*, è la chiave primaria di tipo (int, autoincrement) ed ha lo scopo di identificare in modo univoco un enzima
- *nome\_enzima*, è la stringa con vincolo di univocità che rappresenta il nome dell'enzima

Non ho messo come chiave primaria il nome dell'enzima in quanto quest'ultimo potrebbe essere modificato da parte dell'utente.

**Effettori:** la tabella *effettori* memorizza il nome di tutti gli effettori con la loro natura (metalli, pesticidi, organici, ...). Ciascun effettore è perciò descritto da tre campi:

- *pk\_effettore*, è la chiave primaria di tipo (int, autoincrement) ed ha lo scopo di identificare in modo univoco un effettore
- *nome\_effettore*, è la stringa con vincolo di univocità che rappresenta il nome dell'effettore
- *tipo\_effettore*, è una stringa che rappresenta la tipologia dell'effettore

**Dati degli esperimenti:** abbiamo quattro tabelle per la memorizzazione dei dati degli esperimenti, una per ciascuna coppia (tipo, concentrazione). Hanno tutte la stessa struttura descritta di seguito:

- *fk\_enzima*, è la chiave esterna che collega un esperimento all'enzima relativo
- *fk\_effettore*, è la chiave esterna che collega un esperimento all'effettore relativo
- *media* e *dev\_standard* sono di tipo real e rappresentano i valori dei risultati degli esperimenti
- *fk\_enzima*, *fk\_effettore* oltre ad essere chiavi esterne, insieme fanno la chiave primaria della tabella

**Reflui:** rappresenta l'entità refluo del modello e raccoglie tutte le informazioni di interesse per caratterizzare in modo più preciso un refluo. Le informazioni di interesse sono:

- *pk\_refluo*, è la chiave primaria (int, autoincrement) che permette l'identificazione univoca del refluo
- *tipo\_esperimento*, è una stringa che può assumere i seguenti valori: "diretti1x", "diretti10x", "incubazione1x" e "incubazione10x" e indica rispetto a quale tipologia di dati sperimentali è stata calcolata la sua composizione del refluo
- *provenienza* è una stringa che rappresenta dove è stato prelevato il refluo in questione
- *data* è una stringa (SQLite non mette a disposizione il tipo *date*) che rappresenta la data in cui il refluo è stato prelevato

**Reflui\_to\_enzimi:** rappresenta il legame tra l'entità refluo e gli enzimi su cui il refluo viene valutato per indurre la sua composizione. I campi di questa tabella sono:

- *fk\_refluo* è la chiave esterna che lega il refluo ad un certo enzima
- *fk\_enzima* è la chiave esterna che lega l'enzima ai suoi dati
- *valore* è di tipo real e rappresenta il livello di attività dell'enzima nel refluo

**Istanza\_risultato:** ogni calcolo della composizione di un refluo genera tre liste di effettori: innocui, singoli e le combinazioni additive. I risultati di tipo “innocui” e “singoli” sono trattati in modo diverso dalle “combinazioni additive”. Per i primi verranno create rispettivamente due entry in questa tabella le quali avranno come valore di “tipo\_risultato” 2 e 3. Per quanto riguarda le combinazioni additive, avremmo una entry per ogni combinazione il cui “tipo\_risultato” avrà valore 1. I campi di questa tabella sono:

- *pk\_istanza* è la chiave primaria di tipo (int, autoincrement)
- *fk\_refluo* è la chiave esterna che lega il risultato dell'elaborazione al corrispondente refluo
- *tipo\_risultato* è di tipo int e rappresenta la tipologia del risultato (additivo, innocuo, singolo)

**Componenti\_istanza:** rappresenta il collegamento tra gli effettori e i risultati del calcolo della composizione di un refluo.

- *fk\_istanza*, è la chiave esterna che si lega alle istanze dei risultati
- *fk\_effettore*, è la chiave esterna lega un effettore ad un refluo

## 4.3 Interfaccia grafica

Per facilitare l'interazione dell'utente finale con il software che calcola la composizione del refluo si è reso necessario lo sviluppo di un'interfaccia grafica. Prima di entrare nel dettaglio, vediamo una panoramica delle funzionalità messe a disposizione dall'interfaccia:

- Gestione completa dei dati: inserimento, modifica, cancellazione e visualizzazione di enzimi, effettori e dati sperimentali.  
Nell'inserimento dei dati sperimentali si è cercato di mantenere la stessa logica con i dati venivano memorizzati in passato: enzima → tipo di esperimento (diretto 1x, diretto 10x, ...)
- Calcolo della composizione dei reflui con gestione dell'archivio delle computazioni passate
- Mediante una mappa è possibile vedere lo stadio degli esperimenti effettuati, ovvero, per ogni enzima nel sistema vengono indicate le tipologie di esperimenti a cui sono stati sottoposti. Ad esempio, osservando la Figura.4.10, vediamo che per l'enzima *gst* abbiamo a disposizione solo esperimenti di tipo diretto eseguiti con ad una concentrazione di 10 volte superiore rispetto ai limiti di legge; questa informazione può essere utile prima di saggiare gli enzimi con un refluo in quanto la scelta degli enzimi sarà influenzata dalla disponibilità dei dati di laboratorio.
- È possibile esportare i dati, sia relativi agli esperimenti che relativi ai reflui e ai risultati del calcolo della loro composizione in formato *.csv* in modo da poter consultare i dati anche al di fuori dell'applicazione, mediante un semplice editor di testo oppure un foglio di calcolo.

Inoltre, per i dati degli esperimenti e dei reflui esiste la possibilità di visualizzare il relativo grafico in *Microsoft Excel*<sup>7</sup>.

Nome enzima	Diretti 1x	Diretti 10x	Incubazione 1x	Incubazione 10x
ada				
chimotripsina				
glucosio_ossidasi				
gst				
laccasi				
ldh				
tripsina				

Figura 4.10: Visualizzazione dello stato di avanzamento dei test di laboratorio.

Le funzionalità offerte da EnzimiLab sono tutte accessibili mediante la barra dei menu. Alcune di queste, quelle che ho ritenuto fossero più frequenti, sono presenti anche come bottone nella toolbar.

### Gli enzimi

La lista degli enzimi, visibile subito all'avvio dell'applicazione, appare nella seconda pagina nel notebook di sinistra. La loro visualizzazione è ordinata lessicograficamente. Per aggiungere o eliminare un enzima si può selezionare dal menu *Enzimi* rispettivamente l'operazione "*Aggiungi un enzima*" e "*Elimina un enzima*" oppure, se ci troviamo sulla pagina degli enzimi, si può compiere tali operazioni selezionando rispettivamente il primo e il secondo bottone nella toolbar, partendo da sinistra. L'aggiunta di un enzima richiede l'inserimento del nome dell'enzima; il nome deve rispettare delle regole: può contenere solo lettere maiuscole, minuscole e il simbolo '\_',

<sup>7</sup>Le versioni di Microsoft Excel supportate sono: XP e 2003.



non può contenere spazi e infine deve iniziare e finire per lettera minuscola<sup>8</sup>. Nel caso il nome inserito non rispetti queste condizioni viene visualizzato un messaggio “Nome non valido” e viene riproposta la finestra di inserimento. Nel caso il nome dell’enzima sia già presente viene comunicato che il nuovo nome non è stato inserito. Per l’eliminazione dell’enzima viene

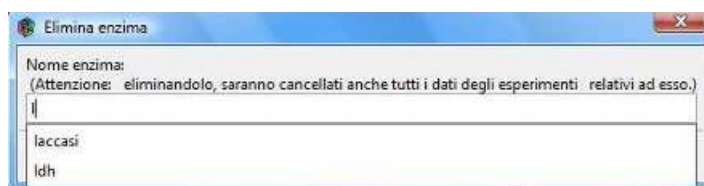


Figura 4.11: Eliminazione di un enzima: durante la digitazione del nome dell’enzima, vengono suggeriti dei candidati.

richiesto il suo nome, suggerendo (vedi Figura. 4.11) possibili nomi di enzimi in relazione alle lettere immesse. Quando un enzima viene eliminato, anche tutti i dati degli esperimenti relativi ad esso vengono eliminati ma non i reflui in cui è stato utilizzato come biomarcatore.

### Gli effettori

L’elenco degli effettori, sempre visibile nella terza pagina del notebook di sinistra, visualizza in ordine alfabetico, rispetto al nome, tutti gli effettori con la loro tipologia (organico, metallo, pesticida, ...). Per aggiungere un effettore è possibile selezionare l’operazione *Aggiungi effettore* dal menu *Effettori* oppure, se la pagina corrente del notebook è sulla lista degli effettori, premere lo stesso bottone utilizzato per l’aggiunta di un enzima, nella toolbar. Il bottone controlla se la pagina corrente nel notebook è quella dell’effettore e in caso positivo farà visualizzare il dialogo per

<sup>8</sup>Queste restrizioni sono dovute all’interfacciamento tra C e Prolog

l'inserimento del nome dell'effettore (sottoposto agli stessi vincoli del nome dell'enzima) e della sua tipologia. L'operazione di eliminazione dell'effettore si trova sempre nel menu degli *Effettori* oppure cliccando sul secondo bottone della toolbar, partendo da sinistra (lo stesso che permette l'eliminazione di un enzima). Anche in questo caso c'è il completamento automatico del nome dell'effettore da eliminare.

### I dati degli esperimenti

**Visualizzazione:** i dati degli esperimenti vengono visualizzati per enzima e tipo di esperimento. Selezionando il menu *Visualizza esperimenti*, occorrerà scegliere la tipologia di esperimento da visualizzare e successivamente l'enzima di cui vogliamo i dati (Figura. 4.12). Quest'ultimi saranno mostrati in una pagina del notebook di destra all'interno di una tabella ordinata lessicograficamente rispetto al nome degli enzimi.

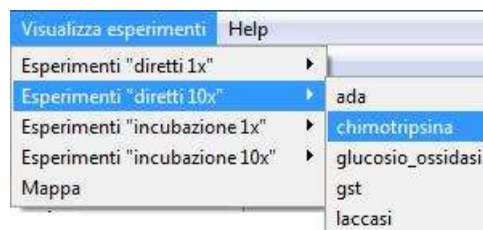


Figura 4.12: Selezione dei dati da visualizzare: occorre scegliere il tipo di esperimento e poi l'enzima.

**Mappa:** la mappa dei dati degli esperimenti è una tabella (Figura. 4.10) visualizza nel notebook di destra, nella quale per ogni enzima viene indicato se ci sono (rettangolino blu) o meno (rettangolino bianco) dei dati sulla sua suscettibilità per ogni tipo di esperimento. Nel caso di

presenza dei dati è possibile cliccare due volte sul rettangolo blu per visualizzarne i dati (come se la visualizzazione dei dati fosse stata selezionata dal menu *Visualizza esperimenti*).

**Inserimento:** per aggiungere i risultati degli esperimenti relativi ad un enzima occorre selezionare il bottone *ampolla*<sup>+</sup> (nella toolbar) e tramite il menu a tendina selezionare l'enzima, la tipologia degli esperimenti e quanti risultati si vuole inserire (se si mette un numero maggiore e non si inseriscono, non succede niente). Nell'esempio visualizzato in Figura. 4.13 vengono inseriti i risultati di quattro esperimenti relativi all'enzima *gst*. Le informazioni richieste per ogni esperimento sono: l'effettore, la media dell'attività dell'enzima e la deviazione standard.

È consigliabile assegnare sempre un valore maggiore di zero alla deviazione standard in quanto un effettore viene selezionato come componente di un refluio se contribuisce alla diminuzione dell'attività enzimatica nella stessa misura rilevata nel refluio: se la deviazione standard è diversa da zero, si può considerare un intervallo di errore  $[media - deviazione\ standard, media + deviazione\ standard]$ , quindi è più probabile che l'attività enzimatica cada in un intervallo piuttosto che sia esattamente identica al valore della media.

Al termine dell'inserimento dei dati verranno visualizzati tutti i dati dell'enzima (del tipo appena inserito) in una pagina del notebook di destra. Nel caso fosse già aperta, verrà aggiornata.

**Modifica ed eliminazione:** la gestione di queste due operazioni segue la stessa logica dell'inserimento di nuovi dati. Si accede a queste operazioni cliccando nella toolbar il bottone *ampolla*<sup>-</sup>. Sarà richiesto



Figura 4.13: Aggiunta dei dati: prima occorre scegliere l'enzima, la tipologia degli esperimenti di cui vogliamo inserire i risultati e quanti ne vogliamo inserire dopodiché verrà visualizzata una finestra per l'inserimento.

il nome dell'enzima e la tipologia di esperimenti di cui si vogliono modificare/eliminare i dati. I dati da modificare verranno visualizzati in una finestra modale, sotto forma di tabella con la seguente struttura: nome effetto, media, deviazione standard, bottone per l'eliminazione della riga, bottone per la modifica della riga. Ogni volta che viene effettuata una modifica o un'eliminazione, la finestra viene aggiornata. Al termine di tutte le operazioni verrà aperta una pagina (o aggiornata) nel notebook di destra per la visualizzazione completa dei dati in questione.

## I reflui

**Visualizzazione reflui:** L'elenco completo dei reflui è visibile, all'avvio dell'applicazione, nella prima pagina del notebook di sinistra. Hanno una struttura ad albero organizzata in relazione alle caratteristiche visualizzate: provenienza, data di prelievo, identificatore univoco, tipologia dei dati utilizzati per il calcolo della sua composizione, enzimi testati e le loro risultanti attività.

Provenienza	Data	Id reflow	Tipo esperimento	Enzimi	Valori
agenziaXY	1/1/2009	2	Incubazione 1x	chimotripsina	92.000000
				tripsina	90.000000
chimiciABC	3/1/2009	1	Incubazione 1x	chimotripsina	67.000000
				tripsina	74.000000
uslPisa	1/1/2009	3	Incubazione 10x	chimotripsina	94.000000

Figura 4.14: La visualizzazione dei reflui come albero.

Per rivedere la composizione del reflow elaborata occorre selezionare dal menu *Reflui* l'operazione "Visualizza risultati reflow" e indicare l'identificatore del reflow di interesse. Nel notebook di destra saranno aperte tre pagine, relative rispettivamente a: le combinazioni additive di effettori, gli effettori singoli e quelli innocui.

**Composizione del reflow:** per calcolare la composizione di un reflow selezionare dal menu *Reflui* o dalla toolbar "Calcola composizione reflui". Si potranno inserire dati caratterizzanti il reflow come la sua provenienza e la data del prelievo; inoltre si dovrà specificare il relazione a quali dati sperimentali saranno calcolati i suoi componenti e rispetto a quali enzimi, indicando il loro livello di attività. Gli enzimi con livello nullo di attività saranno ignorati. Al termine dell'elaborazione (Figura. 4.15) nel notebook di destra verranno aperte tre pagine relative rispettivamente a: combinazioni additive di effettori, lista degli effettori innocui e lista degli effettori singoli.

Inoltre verrà aggiornata la pagina in cui i reflui sono visualizzati con una struttura ad albero (la prima pagina del notebook di sinistra) con i dati relativi al nuovo reflu.

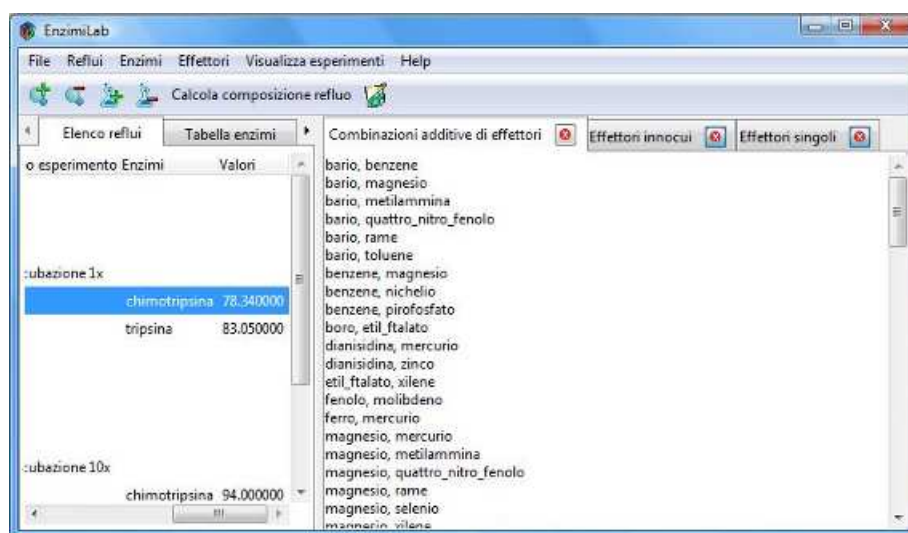


Figura 4.15: Calcolo della composizione del refluo: nel notebook di sinistra sono evidenziati i dati del refluo di cui è appena stata calcolata la composizione mentre nel notebook di destra si vedono le tre pagine dei risultati e in primo piano vediamo le combinazioni additive di effettori.

**Eliminazione del refluo:** per effettuare questa operazione selezionare all'interno del menu *Reflui* l'operazione "Elimina refluo". Per portare a termine l'azione viene richiesto l'identificatore del refluo che siamo intenzionati ad elidere. Tale identificatore è visualizzato nella struttura ad albero dove sono elencati tutti i reflui. È inoltre possibile eliminare tutti i reflui selezionando la checkbox (Figura. 4.16).

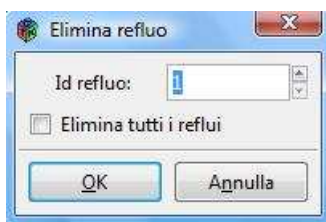


Figura 4.16: Eliminazione reflui: si può eliminare un reflu inserendo il suo identificativo, visibile nella struttura ad albero, oppure selezionando la checkbox si eliminano tutti.



Figura 4.17: Esportazione dei dati: si posso esportare i dati degli esperimenti, dei reflui e della loro composizione. A parte per quest'ultimi che si possono esportare solo nel formato ".csv", gli altri hanno la possibilità di essere esportati con Microsoft Excel.

### Esportazione dei dati

Tutti i dati gestiti da EnzimiLab, sia quelli introdotti dall'utente riguardanti gli esperimenti sia quelli prodotti dall'applicazione stessa mediante l'elaborazione della composizione dei reflui, possono essere esportati. Questa operazione si trova all'interno del menu *File* sotto la voce "Esporta" (Figura. 4.17)

I dati risultanti dall'elaborazione della composizione del reflu al momento possono essere semplicemente esportati in *.csv* in quanto non abbiamo ancora un'idea grafica precisa che li rappresenti mentre i dati degli

esperimenti hanno due possibilità di esportazione:

- In formato *.csv* il quale permette di accedere ai dati semplicemente con un editor di testo oppure con un foglio di calcolo
- Creazione diretta di un file *.xls* in cui vengono inseriti i dati e viene disegnato il relativo grafico, come in Figura. 4.2. Attualmente per usufruire di questo tipo di esportazione è necessario aver installato sul sistema almeno *Microsoft Excel XP*.

#### 4.3.1 Collegamento con l'applicazione Prolog

Come introdotto precedentemente l'algoritmo che calcola la composizione del refluo è scritto in Prolog (*refluo.pl*) ed è un modulo indipendente dall'interfaccia: per integrarlo all'interno di quest'ultima è stato necessario creare un programma scritto C (*refluo-c.c*) che comunicasse i dati del calcolo al Prolog e interpretasse i suoi risultati.

Il SICStus Prolog a questo fine fornisce un'interfaccia bi-direzionale composta da un insieme di funzioni e macro per varie operazioni[25].

##### Come chiamare il Prolog dal C

Vediamo le funzioni dell'interfaccia utilizzate da *refluo-c.c* per comunicare con il Prolog:

- La prima operazione da fare è inizializzare il runtime del Prolog. A questo fine si utilizza la funzione *SP\_initialize()* la quale alloca un'area di memoria in cui il Prolog può gestire i propri dati e caricare le librerie di cui ha bisogno. Questa funzione ritorna *SP\_SUCCESS* se l'inizializzazione va a buon fine, altrimenti *SP\_ERROR*.



- Tramite le funzioni *SP\_restore* e *SP\_load* è possibile ripristinare l'ambiente per l'esecuzione del programma Prolog e caricare i file esterni da cui il programma va a leggere i dati.

```
...

/* Nomi dei file principali*/
#define FILE_ENZIMI      "enzimi.pl"
#define FILE_EFFETTORI  "effettori.pl"
#define FILE_RESTORE    "teoria_refluo.sav"

/* Nomi dei file in cui ci sono i dati degli esperimenti */
#define FILE_DIRETTI1X   "dati_diretti1x.pl"
...

/* Tipologie di esperimenti */
#define DIRETTI1X        1

...

int rval = SP_restore(FILE_RESTORE);

if (rval == SP_ERROR || rval == SP_FAILURE)
{
    fprintf(stderr, "Could not restore \"%s\".\n", FILE_RESTORE);
    exit(101);
}

rval = -1;
rval = SP_load(FILE_ENZIMI);
if (rval == SP_ERROR || rval == SP_FAILURE)
{
    fprintf(stderr, "Could not load \"%s\".\n", FILE_ENZIMI);
    exit(102);
}

rval = -1;
rval = SP_load(FILE_EFFETTORI);
if (rval == SP_ERROR || rval == SP_FAILURE)
{
    fprintf(stderr, "Could not load \"%s\".\n", FILE_EFFETTORI);
```

```
        exit(103);
    }

    ...

    /* tipo_esperimento e' un intero passato come argomento che indica quali
       dati dover caricare
    */
    switch(tipo_esperimento)
    {
        case DIRETTI1X:
            rval = SP_load(FILE_DIRETTI1X);
            break;

        ...
    }

    if (rval == SP_ERROR || rval == SP_FAILURE)
    {
        switch(tipo_esperimento)
        {
            case DIRETTI1X:
                fprintf(stderr, "Could not load \"%s\".\n", FILE_DIRETTI1X);
                exit(104);
                break;

            ...
        }
    }
}
```

- *Per poter utilizzare un predicato dal C* occorre cercare la sua definizione rispetto a: modulo di appartenenza, nome del predicato e sua arietà.

La funzione *SP\_predicate()* ritornerà un puntatore a questa definizione oppure NULL se il predicato non è presente nel modulo indicato.

L'inserimento del nome del modulo è opzionale, quindi se il suo valore è NULL oppure stringa vuota il modulo assunto come default sarà

*user.*

```

/* Creo l'argomento del predicato che rappresenta il nome dell'enzima */
input_enzima = SP_new_term_ref();
SP_put_string(input_enzima, argv[i]);

/* Creo l'argomento del predicato che rappresenta il valore
dell'attivita' dell'enzima nel reflu
*/
double valore_enzima = atof(argv[i+1]);
attivita_nel_refluo = SP_new_term_ref();
SP_put_float(attivita_nel_refluo, valore_enzima);

// Cerco la definizione del predicato 'elabora_coppia/2'.
if (!(pred = SP_predicate("elabora_coppia", 2, NULL)))
{
    fprintf(stderr, "Could not find elabora_coppia/2.\n");
    exit(108);
}

```

- *Per poter chiamare un predicato* si utilizza la funzione *SP\_query()*, la quale crea un goal sulla base della definizione del predicato (presa come descritto dal passo precedente) e dei suoi argomenti. I valori ritornati dalla funzione sono *SP\_SUCCESS* nel caso il goal abbia successo, *SP\_FAILURE* se fallisce e *SP\_ERROR* se si presenta una condizione di errore.

```

goal = SP_query(pred, input_enzima, attivita_nel_refluo);
if(goal == SP_SUCCESS)
{
    /* E' un predicato dinamico, se il goal va a buon fine significa che il predicato
    e' stato aggiunto
    */
    ...
}
else if(goal == SP_FAILURE || goal == SP_ERROR)
{ ... }

```

- Gli argomenti di un predicato possono essere inizializzati con un valore oppure possono essere variabili in cui il prolog andrà a mettere un valore. L'inizializzazione degli argomenti di un predicato l'abbiamo vista nell'esempio della chiamata del predicato *elabora\_coppia/2*. Gli argomenti del predicato utilizzati come variabili vengono utilizzati quando richiediamo il calcolo della composizione del reflu; in questo caso il Prolog metterà negli argomenti la soluzione calcolata.

```
/* Crea i tre argomenti del predicato 'effettori_nel_refluo/3'. */
SP_put_variable( effettori_singoli = SP_new_term_ref());
SP_put_variable( effettori_innocui = SP_new_term_ref());
SP_put_variable( effettori_additivi = SP_new_term_ref());

/* Esecuzione del goal '|?- effettori_nel_refluo(S, I, A).' */
if (!(goal = SP_query(pred, effettori_singoli, effettori_innocui, effettori_additivi)))
{
    fprintf(stderr, "Failed query effettori_nel_refluo/3.\n");
    exit(113);
}
```

- Per prelevare il valore restituito dal Prolog da un termine ci sono diverse funzioni, a seconda del valore ritornato[26]. Nel nostro caso il Prolog ci restituisce tre liste di stringhe, perciò le funzioni che abbiamo utilizzato sono *SP\_get\_list* e *SP\_get\_string*.

```
...

if(SP_is_list(effettori) != 0)
{
    ...
    SP_term_ref head = SP_new_term_ref();
    SP_term_ref tail = SP_new_term_ref();

    SP_get_list(effettori,head,tail);
    SP_get_string(head,&text);

    ...
}
```

```
fprintf(fd_filename, "%s ", text);  
  
...  
}
```

### Creazione dell'eseguibile

Vediamo le principali operazioni da svolgere per costruire un “*All-in-one Executable*” che sarà poi chiamato all'interno dell'interfaccia utente:

1. Creazione del file “.sav” contenente il codice compilato di *refluo.pl* e di tutte le librerie Prolog utilizzate. La creazione di questo file viene effettuata mediante il prompt dell'interprete Prolog:

```
SICStus 3.11.2 ...  
Licensed to di.unipi.it  
| ?- compile(refluo.pl).  
% compiling .../refluo.pl...  
% ...loading several library modules  
| ?- save_program('refluo.sav').  
% .../refluo.sav created in 201 msec  
| ?- halt.
```

2. La creazione del file “.exe” di *refluo\_c.c* da linea di comando va effettuata all'interno di “*Visual Studio Command Prompt*”, in quanto la versione di SICStus per Windows prevede come compilatore per il linguaggio C “cl”.

```
C:\Users\elisa\reflui> spld --main=user reflu_c.c -o reflu_c.exe
```

*spld* è il comando del SICStus Prolog che permette la creazione di un eseguibile stand-alone. Con “--main=user” indichiamo al comando che la funzione *main* del programma C si chiama *user\_main*; gli altri due parametri sono il nome del sorgente C e il nome del file di output.

Se l'esecuzione del comando va a buon fine, saremo in grado di calcolare la composizione di un refluo direttamente da linea di comando. Nell'esempio seguente viene richiesto il calcolo della composizione di un refluo, rispetto ai dati sperimentali diretti eseguiti alla concentrazione di legge (direttlx) in cui l'enzima *ldh* ha un livello di attività pari al 98.34%:

```
C:\Users\elisa\reflui> .\refluo_c.exe 1 ldh 98.34
**** Lista combinazioni additive ****
argento, bario
argento, cadmio
....
*****
**** Lista singoli ****
argento, fluoro, mercurio, rame ...
*****
**** Lista innocui ***
ferro, zinco, benzene, etil_ftalato ...
*****
```

3. All'interno dell'interfaccia grafica l'eseguibile creato al passo precedente viene lanciato attraverso la chiamata di sistema di Windows *spawnv(mode, path, args)*<sup>9</sup> che permette di eseguire il programma indicato come secondo argomento in un altro processo. Se l'argomento *mode* è uguale a *\_P\_WAIT*<sup>10</sup> il processo che esegue la chiamata resta in attesa della terminazione del processo figlio: la funzione restituisce il pid del processo se questo termina normalmente, altrimenti il segnale che ha ucciso il processo. L'ultimo

---

<sup>9</sup>Il corrispettivo sotto linux è: fork seguita da una *execv*.

<sup>10</sup>Possibili valori di *mode*: *P\_NOWAIT*, *P\_NOWAITO*, *P\_WAIT*, *P\_DETACH*, *P\_OVERLAY*. Per un approfondimento visitare il sito: [http://msdn.microsoft.com/en-us/library/20y988d2\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/20y988d2(VS.71).aspx)

argomento, *args*, è la lista dei parametri che saranno letti dal main del processo figlio.

### 4.3.2 Collegamento con Microsoft Excel

Il modulo per l'esportazione dei dati in Microsoft Excel è stato sviluppato in C# all'interno di *Microsoft Visual Studio 2008* sfruttando le librerie "Microsoft. Office. Interop. Excel" le quali permettono in modo semplice di creare fogli di calcolo e grafici formattandoli a nostro piacimento sfruttando il paradigma ad oggetti.

Principalmente, il modello a oggetti emula direttamente l'interfaccia utente. L'oggetto *Application*, ad esempio, rappresenta l'intera applicazione e ogni oggetto *Workbook* (il contenitore dei fogli di calcolo) contiene un insieme di oggetti *Worksheet* (foglio di calcolo). Al livello immediatamente inferiore, la principale astrazione utilizzabile per rappresentare le celle è costituita dall'oggetto *Range*, che consente di utilizzare singole celle o gruppi di celle. L'oggetto che permette la creazione dei grafici è *Chart*, il quale mette a disposizione tutti i tipi di grafici che si possono trovare all'interno di Microsoft Excel; noi abbiamo usato *XlChartType.xlRadar*. Dall'interno di EnzimiLab viene chiamato l'eseguibile "ExcelExporter" che accedendo al database dell'applicazione crea un foglio di calcolo con il relativo grafico. Richiede due parametri: il nome dell'enzima o l'identificatore del refluio e un intero che indica se vanno prelevati i dati relativi agli esperimenti di un enzima o invece gli enzimi su cui era stato testato il refluio. Al termine dell'elaborazione dei dati verrà avviato Microsoft Excel il quale mostrerà:

**Per i reflui** due pagine: nella prima saranno elencati gli enzimi con i valori della loro attività catalitica mentre nella seconda sarà

visualizzato il grafico.

**Per l'enzima** due pagine per ogni tipologia di esperimento (quindi otto in totale) dove nella prima pagina saranno visibili gli effettori con la media e la deviazione standard e nella seconda tanti grafici quante sono le tipologie degli effettori (questo per garantire una maggiore chiarezza nel grafico nel caso in cui gli effettori siano molti).



# Capitolo 5

## Conclusioni e sviluppi futuri

Lo scopo di questo progetto è stato formalizzare il dominio dei reflui e dare delle indicazioni sulla loro possibile composizione. Inoltre siamo riusciti a dare un'organizzazione uniforme al grande ammontare di dati creati dai biologi e uno strumento semplice per il loro mantenimento in quanto tramite l'interfaccia possono inserire, modificare, cancellare ed esportare i dati degli esperimenti e dei reflui testati in .csv e .xls.

La stima della composizione del refluo attualmente viene fatta sfruttando l'idea di *additività* come semplice addizione degli effetti delle sostanze tossiche sull'attività enzimatica a causa dell'assenza di conoscenze più approfondite sugli effetti congiunti di queste sostanze sugli enzimi selezionati.

Se in futuro avremo a disposizione informazioni aggiuntive non solo riguardanti gli effetti congiunti degli effettori sull'attività catalitiche di un enzima, ma anche informazioni riguardanti la probabilità con la quale un determinato insieme di effettori può stare insieme potremmo inserire dei vincoli sulla generazione delle combinazioni all'interno del nostro programma Prolog in modo da renderlo più preciso. La generazione delle

combinazioni di effettori sarebbe enormemente avvantaggiata da queste informazioni in quanto se ad esempio viene dimostrato che *ferro* e *magnesio* non si possono mai trovare insieme potremmo evitare di costruire le combinazioni con questi effettori.

Grazie all'indipendenza del nucleo, il programma Prolog, dal resto del sistema queste modifiche sarebbero semplici e veloci.

In futuro sarà possibile aggiungere funzionalità per analizzare la frequenza con cui gli effettori sono presenti nei reflui, in relazione alla provenienza o ad altri fattori caratterizzanti ogni singolo reflu o oppure altri tipi di analisi statistiche opportunamente scelte in relazione ai dati prodotti tramite il calcolo della composizione dei reflui testati.

# Ringraziamenti

Lo sviluppo di questo progetto mi ha entusiasmata e mi ha dato la possibilità di studiare ed approfondire argomenti che altrimenti non avrei mai visto perciò ringrazio i professori *Paolo Mancarella* e *Roberto Barbuti* che ne sono i fautori insieme ai professori *Umberto Mura* e *Antonella del Corso* e alla dottoressa *Elisa Di Bugno* del dipartimento di Biologia di Pisa. Ringrazio il mio datore di lavoro *Daniele Rebecchi* per la flessibilità sugli orari di ufficio permettendomi di gestire al meglio lo studio e i miei *colleghi* che hanno assecondato i miei stati d'animo: ormai riescono a capire il mio umore dal *buongiorno*!

Come potrei non ringraziare tutti i miei amici? In particolare *Arianna, Laura, Speranza, Dera, Marta, Sargon, Peppe, Cristian, Filippo, Matteo e Paolo* per le partite di calcetto, le grigliate, le serate cinematografiche ma soprattutto per le serate passate a giocare a *tronco*!...dobbiamo ancora gustare il buonissimo vino *Ferrucci*...

Vorrei ringraziare *Daniela, Massimo, Giacomo, Marianna, Emanuele ed Agnese* per i momenti spassosi che mi hanno regalato giocando a Burraco e un ringraziamento a *Bruna* per la mitica tombola del Natale...aspetto il Natale solo per quella!

Un abbraccio forte forte a *Michela* per...tutto! L'elenco sarebbe davvero lungo. Grazie!

Non bastano mille ringraziamenti per i *miei genitori* che hanno dedicato la loro vita, specialmente gli anni più belli, a me. Ve ne sarò per sempre grata!

Un bacio a mia nonna *Assuntina* per avermi sopportato anche quando ero una peste, per le partite a carte e i bomboloni alle sagre estive.

Infine ringrazio l'università perché mi ha fatto conoscere una persona meravigliosa che mi è stata vicina per tutto il corso degli studi, mi ha

insegnato a credere in me stessa, ha sopportato tutte le mie paranoie e i miei smarrimenti ed ha condiviso con me i momenti di felicità come oggi. Grazie *Daniele*! Grazie, grazie, grazie...

Speravate fosse finito tutto qua e invece devo ancora ricordare una persona, la più importante, la quale ha lasciato un buco enorme dentro me ma che mi ha regalato tanto amore e mi è stata vicina finché ha potuto. Ogni mattina mi portava all'università e ogni sera tardi mi veniva a riprendere all'aula studio, si ricordava ogni esame e si accorgeva di com'era andata la giornata nel momento in cui entravo in macchina. *Nonno*, questo traguardo lo dedico a te, spero tu sia soddisfatto della tua nipotina... ti voglio bene!

# Bibliografia

- [1] J. Setubal, J. Meidanis: *Introduction to computational molecular biology*, PWS Publishing Company 1997, cap.1
- [2] M. Attimonelli, G. Pesole, E. Quagliariello, C. Saccone: *Principi di Bioinformatica*, Guido Gnocchi Editore 1997, cap.3
- [3] C. Priami: Informatica e biologia dei sistemi, Mondo Digitale, n. 1 - marzo 2004
- [4] J. M. Berg, J. L. Tymoczko, L. Stryer: *Biochimica*, 5 edizione Zanichelli 2003, cap.8
- [5] F. Sartori: *Bioindicatori ambientali*, Fondazione Lombardia per l'Ambiente 1998.
- [6] F. La Rocca: *Biomarcatori e bioindicatori nella valutazione della qualità delle acque interne*, Tesi di Laurea, 2004-2005
- [7] A. Binelli, F. Ricciardi, A. Provini: *Colinesterasi, vecchi biomarker con un nuovo futuro? Possibile impiego nei bivalvi d'acqua dolce.*, Dipartimento di Biologia, Università degli Studi di Milano  
([www.dsa.unipr.it/site/pubblicazioni/attiXII/content/b4.pdf](http://www.dsa.unipr.it/site/pubblicazioni/attiXII/content/b4.pdf))

- 
- [8] I. Cunha, E. Mangas-Ramirez, L. Guilhermino: *Effects of copper and cadmium on cholinesterase and glutathione S-transferase activities of two marine gastropods (Monodonta lineata and Nucella lapillus)*, Comparative Biochemistry and Physiology Part C: Toxicology & Pharmacology, Volume 145, Maggio 2007, Pagine 648-657
- [9] M. J. E. Sternberg, R. A. Lewis, R. D. King, S. Muggleton: *Modelling the structure and function of enzymes by machine learning*, Faraday Discussions:93, Pagine 269-280, 1992
- [10] R. Bergaglio, C. Marchiol, R. Giannantonio: *Biografia di John von Neumann*,  
<http://www.minerva.unito.it/SIS/VonNeumann/VonNeumann01.htm#Indice>  
generale
- [11] T. E. Headrick, B. G. Buchanan: *Some Speculations about Artificial Intelligence and Legal Reasoning*, 1971.
- [12] R. A. Kowalski: *Legislation as Logic Programs*, 1992
- [13] M. Turcotte, S. H. Muggleton, M. J. E. Sternberg: *A Prolog Database Unifying Taxonomic and Structural Information*, Intelligent Systems for Molecular Biology Book of Abstracts, 1998
- [14] C. J. Rawlings, W. R. Taylor, J. Nyakairu, J. Fox, M. J. E. Stenberg: *Using Prolog to represent and reason about protein structure*, Springer-Verlag New York, 1986, Pagine 536-543
- [15] O. Ray, C. H. Bryant: *Inferring the function of genes from synthetic lethal mutations*, 2nd International Conference on Complex, Intelligent and Software Intensive Systems, Febbraio 2008, Pagine 667-671

- 
- [16] L. Calzone, F. Fages, S. Soliman: *BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge*, BioInformatics:22, Pagine 1805-1807, 2006
- [17] C. Bodei, A. Bracciali, D. Chiarugi: *On deducing causality in metabolic networks*, BioInformatics, 2008
- [18] J. W. Lloyd: *Fondamenti di programmazione logica*, Franco muzzio editore, 1986
- [19] I. Bratko: *PROLOG, Programming for Artificial Intelligence* third edition, Addison Wesley 2001
- [20] Y. Castelfranchi, O. Stock: *Macchine come noi, la scommessa dell'intelligenza artificiale*, Editori Laterza, 2003
- [21] L. Sterling, E. Shapiro: *The Art of Prolog*, Biblioteca Scientifica Hoepli, 1994
- [22] S. J. Russell, P. Norving: *Intelligenza Artificiale, un approccio moderno*, UTET libreria, 1998
- [23] E. Lamma: *Warren Abstract Machine: La Realizzazione di Prolog come Macchina Astratta*, Dipartimento di Ingegneria Università di Ferrara
- [24] Swedish Institute of Computer Science: *SICStus Prolog User's Manual*, (<http://www.sics.se/sicstus/>)
- [25] Swedish Institute of Computer Science: *Capitolo 9, Mixing C/C++ and Prolog*, SICStus Prolog User's Manual
- [26] Swedish Institute of Computer Science: *Accessing Prolog Terms*, SICStus Prolog User's Manual, pagine 234-237

- [27] SQLite Documentation: <http://www.sqlite.org/docs.html>